

Near-linear constructions of exact unitary 2-designs

Richard Cleve^{*†}Debbie Leung^{‡†}Li Liu^{*}Chunhao Wang^{*}

Abstract

A unitary 2-design can be viewed as a quantum analogue of a 2-universal hash function: it is indistinguishable from a truly random unitary by any procedure that queries it twice. We show that exact unitary 2-designs on n qubits can be implemented by quantum circuits consisting of $\tilde{O}(n)$ elementary gates in logarithmic depth. This is essentially a quadratic improvement in size (and in width times depth) over all previous implementations that are exact or approximate (for sufficiently strong approximations).

1 Introduction

The uniform distribution on the group consisting of all unitary operations acting on n qubits is captured by the *Haar measure*, which is the unique measure that is invariant under left and right multiplication by any group element. Haar-random unitaries, by their symmetries, facilitate many analyses in quantum information [20, 21, 22, 23, 24]. However, Haar-random unitaries have very high computational complexity, in that most of them cannot be efficiently implemented or reasonably well approximated by circuits of size polynomial in the number of qubits. They require many bits to describe. They also require a lot of randomness to sample.

Unitary 2-designs are probability distributions on finite subsets of the unitary group that have some specific properties in common with the Haar measure. Several common definitions for unitary 2-designs have been proposed and studied, each revolving around a specific property or application, and appropriate notion of approximation [12, 10, 19]. These 2-designs are computable by polynomial-size circuits with short specifications and low sampling complexity.

We focus on *exact* unitary 2-designs. In the exact case, we will see that several commonly used definitions can be shown to be equivalent to each other. One particularly natural definition is that they are *two-query indistinguishable* from Haar-random unitaries. Imagine a game where, at the flip of a coin, U is sampled either according to the Haar measure or with respect to the unitary 2-design. A two-query distinguishing procedure can make two queries to U (each being either in the forward direction as U or in the reverse direction as U^\dagger) as well as other quantum operations that do not depend on U and then outputs a bit. A unitary 2-design has the property that no two-query distinguishing procedure can distinguish between the Haar-random case and the 2-design case with probability greater than $1/2$. By this definition, a unitary 2-design is a quantum analogue of a 2-universal hash function [5] (or, more precisely, 2-universal hash permutation). We will show in Section 2 that this definition is equivalent to previous definitions, including those based on *bilateral twirling* [12] and *channel twirling* [10].

^{*}Institute for Quantum Computing and School of Computer Science, University of Waterloo

[†]Canadian Institute for Advanced Research

[‡]Institute for Quantum Computing and Department of Combinatorics and Optimization, University of Waterloo

1.1 Previous work

The uniform distribution on the Clifford group has been shown to be an exact unitary 2-design in the sense of bilateral twirling [12] and channel twirling [10]. This implies that the circuit complexity is $O(n^2/\log n)$ where the gates are one- and two-qubit gates from the Clifford group [1]. Moreover, the sampling cost is $O(n^2)$ random bits of entropy. In the context of bilateral twirling, [19] shows that a certain process of random circuit generation (introduced in [13]) yields $\varepsilon_{\text{bilateral}}$ -approximate unitary 2-designs of size $O(n(n + \log 1/\varepsilon_{\text{bilateral}}))$, where $\varepsilon_{\text{bilateral}}$ measures the distance of the resulting operation from the ideal one.

Another construction [10] yields circuits of size $O(n \log 1/\varepsilon_{\text{channel}})$ for a notion of approximation that is natural for channel twirling; however, it has been pointed out that this notion of approximation could (at least conceivably) incur a blow-up by a factor that is exponential in n in the bilateral twirl context (see, e.g., Section 2 of [19] and Section 1.1 of [3] for discussion about this). For the more general setting, as far as we know, we might need $\varepsilon_{\text{channel}} \leq \varepsilon_{\text{bilateral}}/2^n$ — so the circuit size becomes $O(n(n + \log 1/\varepsilon_{\text{bilateral}}))$.

For exact unitary 2-designs as well as approximations of them related to bilateral twirling, all of the above constructions incur circuits of size $\tilde{\Omega}(n^2)$ and require $\Omega(n^2)$ random bits of entropy.

Reference [6] proves that there exists a small subgroup of the Clifford group that gives rise to an exact unitary 2-design that uses approximately $5n$ random bits of entropy. However, the circuit complexity for this construction is unknown, beyond the $O(n^2/\log n)$ bound that holds for any Clifford operation. References [18] and [32] study the necessary and sufficient entropy for exact and approximate unitary 2-designs. Approximately $4n$ random bits of entropy are necessary.

1.2 New results

We give three constructions of *exact* unitary 2-designs on n qubits that have the following quantum gate costs (number of one- and two-qubit gates):

- $O(n \log n \log \log n)$ gates (all Clifford gates) for infinitely many n , assuming the extended Riemann Hypothesis is true.
- $O(n \log n \log \log n)$ gates (including non-Clifford gates) for all n , unconditionally.
- $O(n \log^2 n \log \log n)$ gates (all Clifford gates) for all n , unconditionally.

The circuits for the first two constructions can be organized so as to perform their computation in $O(\log n)$ depth; the third in $O(\log^2 n)$ -depth (using the fact that efficient multiplication/convolution algorithms require only $O(\log n)$ -depth [33]). These results are near optimal – in Appendix E, we show that for any unitary 2-design (exact or approximate under Definition 2 or 3), a high probability set of the unitaries have size $\Omega(n)$ and depth $\Omega(\log n)$.

All three constructions above use $5n$ bits of randomness (more precisely, they sample from a uniform distribution on a set of size $2^{5n} - 2^{3n}$). They all consist of unitaries in the Clifford group (even in the second construction, non-Clifford gates are used to compute Clifford unitaries efficiently). The circuits use $\tilde{O}(n)$ ancilla qubits (where each ancilla qubit is initially in state $|0\rangle$ and is restored to this state at the end of the computation). Finally, the cost of the classical process that samples these unitary 2-designs (outputs a description of the quantum circuit) is polynomial in n . The cost is dominated by the complexity of computing square roots in the finite field $\text{GF}(2^n)$.

It should also be noted that our Definition 2 is a new characterization of unitary 2-designs in terms of 2-query indistinguishability that may be of independent interest.

1.3 Significance of the new constructions

Since our constructions yield exact unitary 2-designs, they are automatically valid for all notions and definitions of approximate unitary 2-designs. Our construction thus achieves the minimum known circuit size, depth, and sampling complexity simultaneously, among both exact and approximate unitary 2-designs.

Exact 2-designs offer other advantages. Besides the original operational applications of bilateral and channel twirling, 2-designs have appeared in second moment analysis. For example, they arise in [21], where results are obtained about the decoupling of two quantum systems and quantum channel capacities. An exact 2-design can be used in a “plug-and-play” manner. For example, there exists an encoding operation in any unitary 2-design that, when concatenated with an appropriate inner code, achieves the quantum channel capacity. Thus, our results *automatically* imply the existence of such encoding circuits of $O(n \log^2 n \log \log n)$ Clifford gates and depth $O(\log^2 n)$.

In some applications such as decoupling, the distance from an exact 2-design is amplified by a dimensional factor that can be exponential in n (for example, Theorem 1 in [35]). Using our exact construction, such error term vanishes exactly, so does the issue of the exponential amplification of errors. Thus our results yield potentially tighter bounds while maintaining a circuit size of $\tilde{O}(n)$.

Prior to our work, [3] constructs a method to generate random circuits of size $O(n \log^2 n)$ and depth $O(\log^3 n)$ that does not give rise to a 2-design, yet achieves decoupling and provides small encoding circuits for quantum error correcting codes [4]. The advantage in their approach is that no ancillas are needed, and the circuit may model some random physical processes. However, the depth is higher, and a substantial amount of analysis is required in the aforementioned references to show that the construction and circuit size indeed achieve the tasks with the desired accuracy. Adapting their construction to other applications may also require additional analysis.

2 Definition of a unitary 2-design

We first discuss several definitions that are equivalent to the concept of unitary 2-designs.

Let \mathbb{U}_N denote the group of $N \times N$ unitary matrices. We are interested in distributions over \mathbb{U}_N . The Haar measure on \mathbb{U}_N is the unique measure on \mathbb{U}_N that is invariant under left and right multiplication by any $U \in \mathbb{U}_N$. We denote the Haar measure by $\mu(U)$. Let $\mathcal{E} = \{p_i, U_i\}_{i=1}^k$ denote a finite ensemble of unitary matrices $U_1, U_2, \dots, U_k \in \mathbb{U}_N$ where $p_i \geq 0$ and $\sum_i p_i = 1$.

Sampling from the Haar measure is a powerful technique in quantum information theory. Sometimes, we use a physical procedure that averages over such random choices of unitary transformation (for example [12, 10]). Some other times, we have a randomized argument, for example, in the proof of quantum channel capacity [11, 21, 29, 34], in which the average performance over all possible unitary encodings is evaluated.

We are interested in contexts in which such sampling from the Haar measure can be replaced by sampling from a finite ensemble $\mathcal{E} = \{p_i, U_i\}_{i=1}^k$ of unitary matrices. This can reduce the required resources such as shared randomness, communication to implement the random unitary, as well as the computational complexity of implementing the randomly chosen unitary. We now discuss

several of these circumstances.

The first context is concerned with the expected value of polynomials of the entries of unitary matrices drawn according to some distribution. This definition is essentially the original definition of unitary 2-design in [10], and is useful for proving results in other contexts.

Definition 1. *We say that \mathcal{E} is degree-2 expectation preserving if, for every polynomial $\gamma(U)$ of degree at most 2 in the matrix elements of U and at most 2 in the complex conjugates of those matrix elements,*

$$\sum_{i=1}^k p_i \gamma(U_i) = \int d\mu(U) \gamma(U). \quad (1)$$

In Eq. (1) and throughout the paper, an integral written without a specific domain is taken over \mathbb{U}_N .

The second context is concerned with distinguishing whether a random sample U is drawn from the Haar measure or from the ensemble \mathcal{E} , when an arbitrary distinguishing circuit is allowed to make a total of at most two queries of U or U^\dagger . The most general circuit \mathcal{C} of this form is depicted in Figure 1.

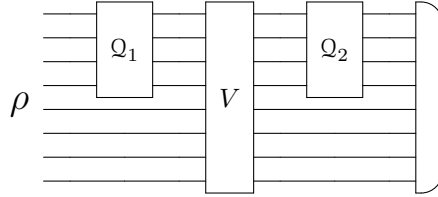


Figure 1: Illustration of a 2-query distinguishing circuit \mathcal{C} . The first query Q_1 can be U or U^\dagger , likewise for the second query Q_2 . The initial state ρ is arbitrary, V is an arbitrary unitary, and the final measurement outputs one bit but is otherwise arbitrary.

The circuit \mathcal{C} starts with an arbitrary initial state ρ (a positive semidefinite matrix of trace 1). Then, the first query, an arbitrary operation V , the second query, and an arbitrary final measurement that outputs one bit are applied in order. We call any such circuit a 2-query distinguishing circuit. If U is drawn from either ensemble, denote the quantum state right before the measurement as $\eta_2(\mathcal{C}, U)$. If U is drawn from \mathcal{E} , the density matrix in \mathcal{C} before the final measurement is $\sum_{i=1}^k p_i \eta_2(\mathcal{C}, U_i)$; similarly, if U is drawn from the Haar measure, the density matrix before the final measurement is $\int d\mu(U) \eta_2(\mathcal{C}, U)$. The output bit of the circuit \mathcal{C} has the same distribution regardless of which ensemble U is sampled from, if and only if the above two density matrices are equal. The following definition describes ensembles that cannot be distinguished by any 2-query distinguishing circuit \mathcal{C} .

Definition 2. *We say that \mathcal{E} is 2-query indistinguishable, if, for any distinguishing circuit \mathcal{C} making up to two queries of a random unitary or its adjoint,*

$$\sum_{i=1}^k p_i \eta_2(\mathcal{C}, U_i) = \int d\mu(U) \eta_2(\mathcal{C}, U). \quad (2)$$

The next context is a special case of the scenario depicted in Figure 1, where U is queried twice in parallel, as illustrated in Figure 2. Consider bipartite operations in which two disjoint systems undergo the same unitary transformation drawn according to some distribution. These

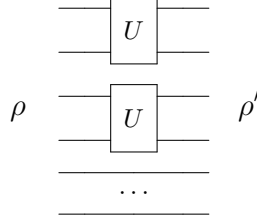


Figure 2: Illustration of the *bilateral twirl*: querying $U \otimes U$. The initial state ρ is arbitrary.

operations are sometimes called bilateral twirls [2, 12]. The \mathcal{E} *bilateral twirl* is defined as the quantum operation

$$\mathcal{T}_{\mathcal{E}}(\rho) = \sum_{i=1}^k p_i (U_i \otimes U_i) \rho (U_i^\dagger \otimes U_i^\dagger). \quad (3)$$

The *full bilateral twirl* is defined as the quantum operation

$$\mathcal{T}_{\mu}(\rho) = \int d\mu(U) (U \otimes U) \rho (U^\dagger \otimes U^\dagger). \quad (4)$$

The full bilateral twirl is motivated operationally [2, 12] and it appears in various mathematical proofs in quantum information [21, 35]. Definition 3 describes ensembles that derandomize the full bilateral twirl.

Definition 3. We say that the ensemble \mathcal{E} implements the full bilateral twirl if $\mathcal{T}_{\mu}(\rho) = \mathcal{T}_{\mathcal{E}}(\rho)$ for all ρ .

The fourth context is concerned with the task of converting any quantum channel into a depolarizing channel of the same average fidelity. This conversion has many important applications, for example, *benchmarking* (for estimating average channel fidelity) of quantum devices [10] and error estimation (for detecting eavesdropping) in quantum key distribution [6].

Let Λ be any quantum channel that maps N -dimensional quantum states to N -dimensional quantum states. An \mathcal{E} -channel-twirl of Λ , denoted by $\mathbb{E}_{\mathcal{E}}(\Lambda)$, is defined as the quantum channel that acts as

$$\mathbb{E}_{\mathcal{E}}(\Lambda) : \rho \mapsto \sum_{i=1}^k p_i U_i^\dagger \Lambda(U_i \rho U_i^\dagger) U_i. \quad (5)$$

In other words, a random change of basis is applied to the system before the channel Λ acts and it is reverted afterwards. A full channel twirl of a quantum channel Λ is given by

$$\mathbb{E}_{\mu}(\Lambda) : \rho \mapsto \int d\mu(U) U^\dagger \Lambda(U \rho U^\dagger) U. \quad (6)$$

Definition 4. We say that \mathcal{E} implements the full channel twirl if $\mathbb{E}_{\mathcal{E}}(\Lambda) = \mathbb{E}_{\mu}(\Lambda)$ for all quantum channels Λ .

Lemma 1 below states that these four relationships between ensembles and the Haar measure are equivalent. Thus, we can think of an ensemble satisfying one of the conditions in alternative ways.

Lemma 1. *Let \mathcal{E} be any ensemble of unitaries in \mathbb{U}_N . Then, the following are equivalent:*

- (1) \mathcal{E} is degree-2 expectation preserving.
- (2) \mathcal{E} is 2-query indistinguishable.
- (3) \mathcal{E} implements the full bilateral twirl.
- (4) \mathcal{E} implements the full channel twirl.

The following corollary of Lemma 1 is not obvious from Definitions 3 and 4 alone.

Corollary 1. *For $\mathcal{E} = \{p_i, U_i\}_{i=1}^k$, let $\mathcal{E}^\dagger := \{p_i, U_i^\dagger\}_{i=1}^k$.*

- (a) \mathcal{E} implements the full bilateral twirl if and only if \mathcal{E}^\dagger does.
- (b) \mathcal{E} implements the full channel twirl if and only if \mathcal{E}^\dagger does.

We note that additional definitions have been discussed in [18, 32, 19, 27]. Several parts of Lemma 1 have been proved in literature [10, 19, 27]. In particular, [27] relates definitions (1), (3), and (4) with bounds on the approximations. We provide a complete (alternative) proof of Lemma 1 and Corollary 1 in Appendix A.

Due to Lemma 1, when we do not need to specify the context, we just call an ensemble satisfying any one of the four conditions a “unitary 2-design.”

3 Pauli mixing implies a unitary 2-design

We describe a simple sufficient condition for \mathcal{E} to be a unitary 2-design.

We begin by reviewing some basic definitions and terminology associated with the Pauli group. Let $X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, $Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$, and $Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$ denote the 2×2 Pauli matrices. For any $a \in \{0, 1\}^n$, define $X^a = X^{a_1} \otimes \cdots \otimes X^{a_n}$ and $Z^a = Z^{a_1} \otimes \cdots \otimes Z^{a_n}$.

Definition 5. *The Pauli group \mathcal{P}_n consists of all operators of the form $i^k X^a Z^b$, where $k \in \{0, 1, 2, 3\}$ and $a, b \in \{0, 1\}^n$. Let $\mathcal{Q}_n = \mathcal{P}_n / \{\pm 1, \pm i\}$, the quotient group that results from disregarding global phases in \mathcal{P}_n (each element of \mathcal{Q}_n can be represented as $P_{a,b} = X^a Z^b$). We call $P_{0,0} = I$ the trivial Pauli.*

Let $H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$ (the 2×2 Hadamard matrix), $S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}$ (the phase gate), and

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \quad (7)$$

Definition 6. *The Clifford group \mathcal{C}_n is the set of all unitary matrices that permute the elements of \mathcal{P}_n (and thus \mathcal{Q}_n) under conjugation.*

The Clifford group \mathcal{C}_n contains the H , CNOT, and S gates, and they form a generating set [17]. Conjugating the elements in \mathcal{P}_n by some $U \in \mathcal{C}_n$ gives a permutation on \mathcal{P}_n ; this also induces a permutation π_U on \mathcal{Q}_n .

Definition 7. Consider an ensemble $\mathcal{E} = \{p_i, U_i\}_{i=1}^k$ of unitary matrices U_1, U_2, \dots, U_k in the Clifford group \mathcal{C}_n . We say that \mathcal{E} is Pauli mixing, if for all $P \in \mathcal{Q}_n$ such that $P \neq I$, the distribution $\{p_i, \pi_{U_i}(P)\}$ is uniform over $\mathcal{Q}_n \setminus \{I\}$.

For any ensemble $\mathcal{E} = \{p_i, U_i\}_{i=1}^k$, let $\mathcal{E}_{\mathcal{Q}} = \{2^{-2n} p_i, U_i R_j\}_{i=1, j=1}^{k, 2^{2n}}$ where R_j ranges over all elements in \mathcal{Q}_n . Intuitively, $\mathcal{E}_{\mathcal{Q}}$ is the ensemble where a random element drawn from \mathcal{E} is preceded by a random Pauli operation drawn from \mathcal{Q}_n .

Pauli mixing by \mathcal{E} is a sufficient condition for the ensemble $\mathcal{E}_{\mathcal{Q}}$ of Clifford unitaries to be a unitary 2-design. More specifically, we have the following lemma.

Lemma 2. Let \mathcal{E} be an ensemble of Clifford unitaries and $\mathcal{E}_{\mathcal{Q}}$ be defined as above. If \mathcal{E} is Pauli mixing, then $\mathcal{E}_{\mathcal{Q}}$ implements the full bilateral twirl.

The original proof of Lemma 2 can be found in [12]. A short proof based on representation theory can be found in [18]. In Appendix D we provide an elementary proof that may be of independent interest. This proof uses some ideas from [12] but has fewer assumptions. In particular, the new proof does not rely on knowing how to evaluate (in closed form) the full bilateral twirl of an arbitrary input state, nor on knowing the invariants of the full bilateral twirl. It is known how to evaluate the full bilateral twirl using representation theory or the double commutant theorem. Our new proof derives this result (how the full bilateral twirl acts) on the side.

Note that, in light of Lemma 1, an alternative way of proving that, whenever \mathcal{E} is Pauli mixing, $\mathcal{E}_{\mathcal{Q}}$ is a unitary 2-design is to use Definition 4. This can be shown in the following two steps. First, conjugating any channel by a uniformly random Pauli operation drawn from \mathcal{Q}_n yields a *mixed-Pauli channel* (a channel that is a probability distribution on the Pauli operators). This is proved in [10]. Second, it is clear that, if \mathcal{E} is Pauli mixing, then conjugating any mixed Pauli channel by a random element of \mathcal{E} results in a depolarizing channel with the same average fidelity as the mixed Pauli channel. This corresponds exactly to an implementation of the full channel twirl.

4 Pauli mixing using the structure of $\text{SL}_2(\text{GF}(2^n))$

For the purposes of analyzing the Clifford group and its action on the Pauli group elements $X^a Z^b = (X^{a_1} \otimes \dots \otimes X^{a_n})(Z^{b_1} \otimes \dots \otimes Z^{b_n})$, it is fruitful to associate a and b with elements of the Galois field of size 2^n . However, for this association to work well technically, we work with two different representations of field elements. If a is represented in some *primal* basis then b is represented in the *dual* of that basis. This section explains this basic framework.

4.1 Review of some properties of Galois fields $\text{GF}(2^n)$

Let $\text{GF}(2^n)$ denote the Galois field of size 2^n (more information about these fields can be found in [28]). The elements of this field form a vector space over $\text{GF}(2)$ so the notion of a basis is well-defined: $\omega_1, \dots, \omega_n \in \text{GF}(2^n)$ are a *basis* if they are linearly independent and span the field; a basis enables us to associate the elements of $\text{GF}(2^n)$ with n -bit strings. A *polynomial basis* of $\text{GF}(2^n)$ is a basis that is of the form $1, \alpha, \alpha^2, \dots, \alpha^{n-1}$, for some $\alpha \in \text{GF}(2^n)$. The standard constructions of $\text{GF}(2^n)$ in terms of irreducible polynomials result in a polynomial basis. However, there are bases that are not necessarily of this form, and these arise in our constructions. For instance, a *normal basis* of $\text{GF}(2^n)$ has the form $\alpha^{2^0}, \alpha^{2^1}, \dots, \alpha^{2^{n-1}}$ for some $\alpha \in \text{GF}(2^n)$.

The dual of a basis is defined in terms of the *trace* function $T : \text{GF}(2^n) \rightarrow \text{GF}(2)$, which is defined as $T(a) = a^{2^0} + a^{2^1} + \dots + a^{2^{n-1}}$. The trace has the property that $T(a+b) = T(a) + T(b)$, for all $a, b \in \text{GF}(2^n)$. In terms of T , we can define the *trace inner product* of $a, b \in \text{GF}(2^n)$ as $T(ab)$. Now, for an arbitrary basis $\omega_1, \dots, \omega_n \in \text{GF}(2^n)$, that we refer to as the *primal basis*, we can define its *dual basis* as the unique $\hat{\omega}_1, \dots, \hat{\omega}_n \in \text{GF}(2^n)$ such that

$$T(\omega_i \hat{\omega}_j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{if } i \neq j. \end{cases} \quad (8)$$

We can associate the elements of $\text{GF}(2^n)$ with n -bit binary strings by taking coordinates with respect to a basis. To facilitate discussion, we use the following notation. With respect to any primal basis $\omega_1, \dots, \omega_n$ and its dual $\hat{\omega}_1, \dots, \hat{\omega}_n$, for $a \in \text{GF}(2^n)$:

- $[a] \in \{0, 1\}^n$ denotes the coordinates of a in the primal basis. Thus, $a = [a]_1 \omega_1 + \dots + [a]_n \omega_n$, which is achieved by setting $[a]_j = T(a \hat{\omega}_j)$ for all $j \in \{1, \dots, n\}$.
- $[a] \in \{0, 1\}^n$ denotes the coordinates of a in the dual basis. Thus, $a = [a]_1 \hat{\omega}_1 + \dots + [a]_n \hat{\omega}_n$, which is achieved by setting $[a]_j = T(a \omega_j)$ for all $j \in \{1, \dots, n\}$.

In some places, where the meaning is clear from the context, it is convenient to write a in place of $[a]$. Also, it is sometimes convenient to think of n -bit binary strings as $\{0, 1\}$ -valued column vectors of length n . Thus, $[a]$ and $[a]$ are sometimes interpreted as binary column vectors of length n . Binary matrices acting on these vectors (in mod 2 arithmetic) are written with square brackets.

It is straightforward to show that the conversion from primal to dual basis coordinates corresponds to multiplication by the $n \times n$ binary matrix

$$W = \begin{bmatrix} T(\omega_1 \omega_1) & \dots & T(\omega_1 \omega_n) \\ \vdots & \ddots & \vdots \\ T(\omega_n \omega_1) & \dots & T(\omega_n \omega_n) \end{bmatrix}. \quad (9)$$

That is, $[a] = W[a]$ (with matrix-vector multiplication in mod 2 arithmetic). Also, $T(ab)$ is the dot-product of the coordinates of a in the primal basis and the coordinates of b in the dual basis:

$$T(ab) = [a] \cdot [b] = [a]_1 [b]_1 + \dots + [a]_n [b]_n \text{ mod } 2. \quad (10)$$

The dual of the dual basis is the primal basis. A basis is *self-dual* if $\omega_i = \hat{\omega}_i$ for all i .

Relative to any basis, multiplication by any particular $r \in \text{GF}(2^n)$ is a linear operator in the following sense. There exists a binary $n \times n$ matrix M_r such that, for all $s \in \text{GF}(2^n)$, $[rs] = M_r[s]$ (with mod 2 arithmetic for the matrix-vector multiplication). In fact, this matrix M_r is

$$M_r = \begin{bmatrix} T(r \hat{\omega}_1 \omega_1) & \dots & T(r \hat{\omega}_1 \omega_n) \\ \vdots & \ddots & \vdots \\ T(r \hat{\omega}_n \omega_1) & \dots & T(r \hat{\omega}_n \omega_n) \end{bmatrix}, \quad (11)$$

and its transpose $(M_r)^T$ corresponds to multiplication by r in the dual basis (that is, $[rs] = (M_r)^T[s]$). It should be noted that algorithms for multiplication in $\text{GF}(2^n)$ are basis dependent; the obvious cost of converting between two bases is $O(n^2)$.

4.2 Pauli mixing from a subgroup isomorphic to $\text{SL}_2(\text{GF}(2^n))$

Due to Lemma 2, it suffices to compute an ensemble of Clifford unitaries that is Pauli mixing. Relative to a (primal) basis, we associate each pair $a, b \in \text{GF}(2^n)$ with the Pauli group element $X^{[a]}Z^{[b]} = (X^{[a]_1} \otimes \dots \otimes X^{[a]_n})(Z^{[b]_1} \otimes \dots \otimes Z^{[b]_n})$. Chau [6] showed that there is a subgroup \mathcal{C} of the Clifford group of size $2^{O(n)}$ such that sampling uniformly over \mathcal{C} performs Pauli mixing. We now give an overview of the approach in [6] (translated into our language). The subgroup \mathcal{C} is isomorphic to the special linear group of 2×2 matrices over $\text{GF}(2^n)$:

$$\text{SL}_2(\text{GF}(2^n)) = \left\{ \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} : \alpha, \beta, \gamma, \delta \in \text{GF}(2^n) \text{ such that } \alpha\delta + \beta\gamma = 1 \right\}.$$

Note that $\text{SL}_2(\text{GF}(2^n))$ has $2^{3n} - 2^n$ elements. The subgroup \mathcal{C} induces a group action of $\text{SL}_2(\text{GF}(2^n))$ on the Paulis by conjugation by certain unitaries.

Definition 8. *With respect to a primal basis for $\text{GF}(2^n)$, we say that a Clifford unitary U induces $M \in \text{SL}_2(\text{GF}(2^n))$ if, for all $a, b \in \text{GF}(2^n)$ and*

$$\begin{pmatrix} a' \\ b' \end{pmatrix} = M \begin{pmatrix} a \\ b \end{pmatrix}, \quad (12)$$

$$UX^{[a]}Z^{[b]}U^\dagger \equiv X^{[a']}Z^{[b']}, \quad (13)$$

where \equiv means equal up to a global phase in $\{1, i, -1, -i\}$ that is a function of M, a , and b .

To rephrase the above definition, suppose $M = \begin{pmatrix} \alpha & \beta \\ \gamma & \delta \end{pmatrix} \in G_0$. Then, for all a, b , conjugating $(X^{[a]_1} \otimes \dots \otimes X^{[a]_n})(Z^{[b]_1} \otimes \dots \otimes Z^{[b]_n})$ by the Clifford unitary U yields $(X^{[\alpha a + \beta b]_1} \otimes \dots \otimes X^{[\alpha a + \beta b]_n})(Z^{[\gamma a + \delta b]_1} \otimes \dots \otimes Z^{[\gamma a + \delta b]_n})$ up to a phase.

We adopt the following notational convention throughout the paper. We write matrices in $\text{SL}_2(\text{GF}(2^n))$ and vectors of length 2 over $\text{GF}(2^n)$ using parenthesis (see above) to distinguish the binary matrices and vectors described in the previous subsection which use square brackets.

It should be noted that, in [6], Eq. (13) is expressed using different notation for the Paulis, that we call *subscripted* Paulis, defined as satisfying $X_a|c\rangle = |a + c\rangle$ and $Z_b|c\rangle = (-1)^{T(bc)}|c\rangle$. It is easy to express these in terms of our *superscripted* Paulis, $X^{[a]}$ and $Z^{[b]}$, as $X_a = X^{[a]}$ and $Z_b = Z^{[b]}$ (since $T(bc) = [b] \cdot [c]$). The occurrence of the dual basis in $Z^{[b]}$ (which is equivalent to using Z_b) in Eq. (13) is not merely a matter of convention: for general $M \in \text{SL}_2(\text{GF}(2^n))$ there *does not exist* a unitary U that induces M in the sense that $UX^{[a]}Z^{[b]}U^\dagger \equiv X^{[a']}Z^{[b']}$. In terms of Definition 8, the following holds.

Lemma 3 ([6]). *With respect to any primal basis of $\text{GF}(2^n)$ and every $M \in \text{SL}_2(\text{GF}(2^n))$, there exists an n -qubit Clifford unitary U that induces M .*

Definition 9. *Consider $M \in \text{SL}_2(\text{GF}(2^n))$. Let U_M denote a unitary that induces M with respect to the primal basis; U_M is unique up to multiplication by a Pauli (a proof of this can be found in [7, 8], and is also provided in Appendix F, Lemma 9). Similarly, let \hat{U}_M denote a unitary that induces M with respect to the dual basis.*

The proof of Lemma 3 in [6] exhibits a possible choice of U_M for any $M \in \text{SL}_2(\text{GF}(2^n))$. However, it is unclear how to implement that U_M as a small quantum circuit, except for the fact

that U_M is in the Clifford group, so its gate complexity is $O(n^2/\log n)$ by [1]. Our results in subsequent sections amount to an alternative proof of Lemma 3 for certain bases of $\text{GF}(2^n)$, as well as a modified version of this lemma. This enables us to ultimately attain gate constructions of size $\tilde{O}(n)$ that implement unitary 2-designs. The relationship between Lemma 3 and unitary 2-designs is based on the fact that the uniform ensemble over $\{U_M : M \in \text{SL}_2(\text{GF}(2^n))\}$ is Pauli mixing, which is a consequence of the following.

Lemma 4 ([6, 18]). *Let G_0 denote the set of all non-zero elements of $\text{GF}(2^n) \times \text{GF}(2^n)$. Let $M \in \text{SL}_2(\text{GF}(2^n))$ be chosen uniformly at random. Then, for any $\begin{pmatrix} a \\ b \end{pmatrix} \in G_0$,*

$$\begin{pmatrix} c \\ d \end{pmatrix} = M \begin{pmatrix} a \\ b \end{pmatrix} \quad (14)$$

is uniformly distributed over G_0 .

Proof. We first show that $\text{SL}_2(\text{GF}(2^n))$ acts transitively on G_0 . Let $\begin{pmatrix} c \\ d \end{pmatrix} \in G_0$. If $c \neq 0$, then, $\begin{pmatrix} c & 0 \\ d & c^{-1} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} c \\ d \end{pmatrix}$. If $c = 0$, then $d \neq 0$, so, $\begin{pmatrix} 0 & d^{-1} \\ d & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} c \\ d \end{pmatrix}$. Thus, we can map any $\begin{pmatrix} c_1 \\ d_1 \end{pmatrix} \in G_0$ to $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and then to any other $\begin{pmatrix} c_2 \\ d_2 \end{pmatrix} \in G_0$ using elements of $\text{SL}_2(\text{GF}(2^n))$.

To prove the lemma, suppose, by contradiction, that there are distinct $\begin{pmatrix} c_1 \\ d_1 \end{pmatrix}, \begin{pmatrix} c_2 \\ d_2 \end{pmatrix}$ such that $\text{Prob}_M \{M \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} c_1 \\ d_1 \end{pmatrix}\} = p_1$, $\text{Prob}_M \{M \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} c_2 \\ d_2 \end{pmatrix}\} = p_2$ and $p_1 > p_2$. But there exists an $M' \in \text{SL}_2(\text{GF}(2^n))$ such that $M' \begin{pmatrix} c_1 \\ d_1 \end{pmatrix} = \begin{pmatrix} c_2 \\ d_2 \end{pmatrix}$. So, $\text{Prob}_M \{M' M \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} c_2 \\ d_2 \end{pmatrix}\} \geq p_1$. But the distribution over M is the same as the distribution over $M' M$, so the left side of the last inequality is p_2 , which is a contradiction. \square

Our goal is to implement U_M (for $M \in \text{SL}_2(\text{GF}(2^n))$) with quantum circuits consisting of $\tilde{O}(n)$ Clifford gates. The interplay between the primal basis and the dual basis is a major complicating factor that we address using two different approaches. In one of our approaches we modify the framework of $\text{SL}_2(\text{GF}(2^n))$.

Our approach in Section 5 is based on a self-dual basis for $\text{GF}(2^n)$ and the structure of $\text{SL}_2(\text{GF}(2^n))$. Our approach in Section 6 is based on a polynomial basis for $\text{GF}(2^n)$ (and its dual) and the structure of two subgroups of $\text{SL}_2(\text{GF}(2^n))$: the lower triangular subgroup and the upper triangular subgroup. These are defined respectively as

$$\Delta_2(\text{GF}(2^n)) = \left\{ \begin{pmatrix} \alpha & 0 \\ \beta & \alpha^{-1} \end{pmatrix} : \alpha, \beta \in \text{GF}(2^n) \text{ and } \alpha \neq 0 \right\} \quad (15)$$

$$\nabla_2(\text{GF}(2^n)) = \left\{ \begin{pmatrix} \alpha & \beta \\ 0 & \alpha^{-1} \end{pmatrix} : \alpha, \beta \in \text{GF}(2^n) \text{ and } \alpha \neq 0 \right\}. \quad (16)$$

These subgroups have interesting mixing properties, albeit weaker ones than $\text{SL}_2(\text{GF}(2^n))$, which are explained in Section 6.

4.3 A framework for implementing elements of $\text{SL}_2(\text{GF}(2^n))$ by unitaries

We first show that all elements of $\text{SL}_2(\text{GF}(2^n))$ can be written as a product of a small constant number of matrices in a generating set—and more restrictive generating sets for $\Delta_2(\text{GF}(2^n))$ and $\nabla_2(\text{GF}(2^n))$. Then we describe Clifford unitaries that induce these generating matrices. In subsequent sections, we show how to implement these unitaries with $\tilde{O}(n)$ quantum gates, thereby implementing elements of $\text{SL}_2(\text{GF}(2^n))$.

Lemma 5. *Every element $M \in \text{SL}_2(\text{GF}(2^n))$ can be expressed as a product of a constant number of the following elements of $\text{SL}_2(\text{GF}(2^n))$:*

$$\begin{pmatrix} r & 0 \\ 0 & r^{-1} \end{pmatrix}, \quad \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}, \quad \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad (17)$$

where $r \in \text{GF}(2^n)$ is non-zero.

Proof. For any $M = \begin{pmatrix} \alpha & \gamma \\ \beta & \delta \end{pmatrix} \in \text{SL}_2(\text{GF}(2^n))$, we can decompose it into a product as follows:

$$\begin{pmatrix} \alpha & \gamma \\ \beta & \delta \end{pmatrix} = \begin{cases} \begin{pmatrix} 1 & 0 \\ \frac{\beta}{\alpha} & 1 \end{pmatrix} \begin{pmatrix} 1 & \alpha\gamma \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \alpha & 0 \\ 0 & \alpha^{-1} \end{pmatrix} & \text{if } \alpha \neq 0 \\ \begin{pmatrix} 1 & 0 \\ \frac{\delta}{\gamma} & 1 \end{pmatrix} \begin{pmatrix} \gamma & 0 \\ 0 & \gamma^{-1} \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} & \text{if } \alpha = 0. \end{cases} \quad (18)$$

Furthermore, for any non-zero $s \in \text{GF}(2^n)$, there exists $t \in \text{GF}(2^n)$ such that $t^2 = s$ (explicitly $t = s^{2^{n-1}}$). This permits us to decompose further as

$$\begin{pmatrix} 1 & 0 \\ s & 1 \end{pmatrix} = \begin{pmatrix} t^{-1} & 0 \\ 0 & t \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} t & 0 \\ 0 & t^{-1} \end{pmatrix} \quad (19)$$

and

$$\begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ s & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \quad (20)$$

□

It is easy to specialize the above lemma to the lower triangular and upper triangular matrices in $\text{SL}_2(\text{GF}(2^n))$ as follows.

Lemma 6. *Every element of $\Delta_2(\text{GF}(2^n))$ can be expressed as a product of a constant number of elements of the form*

$$\begin{pmatrix} r & 0 \\ 0 & r^{-1} \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix} \quad (21)$$

and every element $\nabla_2(\text{GF}(2^n))$ can be expressed as a product of a constant number of elements of the form

$$\begin{pmatrix} r & 0 \\ 0 & r^{-1} \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \quad (22)$$

where $r \in \text{GF}(2^n)$ is non-zero.

In view of Lemma 5, for every M in $\text{SL}_2(\text{GF}(2^n))$, we can find a unitary that induces M if we find a unitary that induces each of $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$, $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$, and $\begin{pmatrix} r & 0 \\ 0 & r^{-1} \end{pmatrix}$ for any non-zero $r \in \text{GF}(2^n)$. Similar statements hold for $\Delta_2(\text{GF}(2^n))$ and $\nabla_2(\text{GF}(2^n))$ with their respectively generating sets shown in Lemma 6.

First consider any non-zero $r \in \text{GF}(2^n)$, and the element of $\text{SL}_2(\text{GF}(2^n))$ of the form

$$\begin{pmatrix} r & 0 \\ 0 & r^{-1} \end{pmatrix}. \quad (23)$$

A Clifford unitary that induces $\begin{pmatrix} r & 0 \\ 0 & r^{-1} \end{pmatrix}$ is the *multiply-by- r* (in the primal basis) operation Π_r defined¹ as $\Pi_r|[\![c]\!]\rangle = |[rc]\rangle$. To improve readability, we henceforth denote $|[\![c]\!]\rangle$ by $|c\rangle$. For example, in this notation, $\Pi_r|c\rangle = |rc\rangle$. Now, to check that Π_r induces $\begin{pmatrix} r & 0 \\ 0 & r^{-1} \end{pmatrix}$, note that, for all $c \in \text{GF}_2(2^n)$,

$$\Pi_r X^{[a]} \Pi_r^\dagger |c\rangle = \Pi_r X^{[a]} |r^{-1}c\rangle \quad (24)$$

$$= \Pi_r |r^{-1}c + a\rangle \quad (25)$$

$$= |c + ra\rangle \quad (26)$$

$$= X^{[ra]} |c\rangle. \quad (27)$$

Furthermore,

$$\Pi_r Z^{[b]} \Pi_r^\dagger |c\rangle = \Pi_r Z^{[b]} |r^{-1}c\rangle \quad (28)$$

$$= \Pi_r (-1)^{[b] \cdot [r^{-1}c]} |r^{-1}c\rangle \quad (29)$$

$$= (-1)^{T(br^{-1}c)} |c\rangle \quad (30)$$

$$= (-1)^{[br^{-1}] \cdot [c]} |c\rangle \quad (31)$$

$$= Z^{[br^{-1}]} |c\rangle. \quad (32)$$

It follows that, for all $a, b \in \text{GF}(2^n)$, $\Pi_r X^{[a]} Z^{[b]} \Pi_r^\dagger = X^{[ra]} Z^{[r^{-1}b]}$. In other words, Π_r induces $\begin{pmatrix} r & 0 \\ 0 & r^{-1} \end{pmatrix}$.

We can write any $\begin{pmatrix} a & b \end{pmatrix} \in \text{SL}_2(\text{GF}(2^n))$ in a primal-dual basis as $\begin{bmatrix} [a] \\ [b] \end{bmatrix} \in \{0, 1\}^{2n}$, where $[a], [b] \in \{0, 1\}^n$. Recall that to distinguish elements of $\text{SL}_2(\text{GF}(2^n))$ from their corresponding binary vectors in a primal-dual basis, we use parenthesis to denote the former and square brackets for the binary vectors and their linear operators.

We summarize the effect of conjugating a Pauli $X^{[a]} Z^{[b]}$ by Π_r on the binary strings $[a]$ and $[b]$ as the following mapping on $2n$ -bit strings:

$$\begin{bmatrix} [a] \\ [b] \end{bmatrix} \mapsto \begin{bmatrix} [ra] \\ [r^{-1}b] \end{bmatrix} = \begin{bmatrix} M_r [a] \\ (M_{r^{-1}})^\top [b] \end{bmatrix}. \quad (33)$$

Here M_r is the linear operator corresponding to multiplication by r in the primal basis, as defined in Eq. (11); $(M_{r^{-1}})^\top$, the transpose of $M_{r^{-1}}$, which (due to the form of Eq. (11)) is the linear operator corresponding to multiplication by r^{-1} in the dual basis.

The following definition is similar to Definition 8.

Definition 10. We say that a Clifford unitary U induces the $2n \times 2n$ binary matrix \mathcal{M} , if, for all n -bit strings s, r ,

$$UX^r Z^s U^\dagger \equiv X^{r'} Z^{s'}, \quad \text{where} \quad \begin{bmatrix} r' \\ s' \end{bmatrix} = \mathcal{M} \begin{bmatrix} r \\ s \end{bmatrix}. \quad (34)$$

¹This unitary operation acts on computation basis states similarly to M_r defined in Sec. 4.1, Eq. (11).

Here, \equiv means equal up to a global phase in $\{1, i, -1, -i\}$ that is a function of \mathcal{M} , r , and s . We also say that U induces the mapping $v \mapsto \mathcal{M}v$.

For example, Π_r induces the mapping given by Eq. (33) and the matrix $\begin{bmatrix} M_r & 0 \\ 0 & (M_{r-1})^\top \end{bmatrix}$.

Now we return to finding unitaries that induce elements of $\text{SL}_2(\text{GF}(2^n))$. Consider the element $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ of $\text{SL}_2(\text{GF}(2^n))$. The Clifford unitary that induces $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ should transform the Pauli $X^{[a]}Z^{[b]}$ along the lines of the mapping

$$\begin{bmatrix} [a] \\ [b] \end{bmatrix} \mapsto \begin{bmatrix} [a] \\ [a+b] \end{bmatrix} = \begin{bmatrix} [a] \\ [a] + [b] \end{bmatrix} = \begin{bmatrix} [a] \\ W[a] + [b] \end{bmatrix} = \begin{bmatrix} I & 0 \\ W & I \end{bmatrix} \begin{bmatrix} [a] \\ [b] \end{bmatrix} \quad (35)$$

where W is the linear operator for primal-to-dual basis conversion defined in Eq. (9).

For *any* symmetric $n \times n$ binary matrix V , we show that there is a diagonal Clifford unitary Γ_V that implements $\begin{bmatrix} I & 0 \\ V & I \end{bmatrix}$. The unitary Γ_V is defined as

$$\Gamma_V |c\rangle = i^{\sum_{j=1}^n \sum_{k=1}^n V_{jk} c_j c_k} |c\rangle. \quad (36)$$

We begin with some preliminary observations. Since, for all $i, j \in \{1, \dots, n\}$, $V_{ij} = V_{ji}$, an equivalent definition is

$$\Gamma_V |c\rangle = i^{\sum_{j=1}^n V_{jj} c_j} (-1)^{\sum_{1 \leq j < k \leq n} V_{jk} c_j c_k} |c\rangle. \quad (37)$$

From Eq. (37), it is clear that Γ_V is in the Clifford group, since it is computed by the following composition of gates: an S gate acting on each qubit j for which $V_{jj} = 1$; and a controlled- Z gate acting on qubits j and k for each $j < k$ where $V_{jk} = 1$ (all these gates commute). This generic construction consists of $O(n^2)$ gates. In Sections 5 and 6, for the primal-to-dual basis conversion matrix W (which is symmetric from its definition in Eq. (9)), we exhibit circuits implementing Γ_W with $\tilde{O}(n)$ gates.

To check that Γ_W induces the mapping in Eq. (35), it is convenient to separately consider the diagonal and off-diagonal entries of W . Let $W = D + E$, where D is diagonal and $E_{jj} = 0$ for all $j \in \{1, \dots, n\}$. This allows us to write $\Gamma_W = \Gamma_{D+E} = \Gamma_D \Gamma_E$, as a direct consequence of

$$\begin{bmatrix} I & 0 \\ W & I \end{bmatrix} = \begin{bmatrix} I & 0 \\ D & I \end{bmatrix} \begin{bmatrix} I & 0 \\ E & I \end{bmatrix}. \quad (38)$$

Then, from the discussion following Eq. (37), we know that $\Gamma_D = S^{W_{11}} \otimes \dots \otimes S^{W_{nn}}$ and it is straightforward to check that

$$\Gamma_D X^{[a]} \Gamma_D^\dagger = \left(i^{W_{11}a_1 + \dots + W_{nn}a_n} \right) X^{[a]} Z^{D[a]}, \quad (39)$$

where we are using the notation $a_i = \lceil a \rceil_i$. For Γ_E , we have

$$\Gamma_E X^{\lceil a \rceil} \Gamma_E^\dagger |c\rangle = \Gamma_E X^{\lceil a \rceil} (-1)^{\sum_{j=1}^n \sum_{k=j+1}^n W_{jk} c_j c_k} |c\rangle \quad (40)$$

$$= \Gamma_E (-1)^{\sum_{j=1}^n \sum_{k=j+1}^n W_{jk} c_j c_k} |a+c\rangle \quad (41)$$

$$= (-1)^{\sum_{j=1}^n \sum_{k=j+1}^n W_{jk} ((a_j+c_j)(a_k+c_k)+c_j c_k)} |a+c\rangle \quad (42)$$

$$= (-1)^{\sum_{j=1}^n \sum_{k=j+1}^n W_{jk} (a_j a_k + a_j c_k + a_k c_j)} |a+c\rangle \quad (43)$$

$$= (-1)^{\sum_{j=1}^n \sum_{k=j+1}^n W_{jk} a_j a_k} (-1)^{\lceil c \rceil \cdot E \lceil a \rceil} |a+c\rangle \quad (44)$$

$$= (-1)^{\sum_{j=1}^n \sum_{k=j+1}^n W_{jk} a_j a_k} X^{\lceil a \rceil} (-1)^{\lceil c \rceil \cdot E \lceil a \rceil} |c\rangle \quad (45)$$

$$= (-1)^{\sum_{j=1}^n \sum_{k=j+1}^n W_{jk} a_j a_k} X^{\lceil a \rceil} Z^{E \lceil a \rceil} |c\rangle. \quad (46)$$

Combining Eqs. (39), (46), and the fact that Γ_W commutes with every $Z^{\lfloor b \rfloor}$, we have

$$\Gamma_W X^{\lceil a \rceil} Z^{\lfloor b \rfloor} \Gamma_W^\dagger = \left(i^{\sum_{j=1}^n \sum_{k=1}^n W_{jk} a_j a_k} \right) X^{\lceil a \rceil} Z^{D \lceil a \rceil + E \lceil a \rceil + \lfloor b \rfloor} \quad (47)$$

$$= \left(i^{\sum_{j=1}^n \sum_{k=1}^n W_{jk} a_j a_k} \right) X^{\lceil a \rceil} Z^{W \lceil a \rceil + \lfloor b \rfloor}, \quad (48)$$

which implies that Γ_W induces the mapping in Eq. (35).

For completeness, a unitary operation that induces the element $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ of $\text{SL}_2(\text{GF}(2^n))$ should also be considered. This is addressed in sections 5 and 6 in very different ways, and we defer the discussion of this to those sections.

5 $\tilde{O}(n)$ implementation based on self-dual basis for $\text{GF}(2^n)$

We want to find $\tilde{O}(n)$ -sized circuits to implement unitaries that induce the generators of $\text{SL}_2(\text{GF}(2^n))$. Our first approach is to represent $\text{GF}(2^n)$ in a self-dual basis. The advantage of using a self-dual basis is that, the change of basis operation W defined in Eq. (9) is simply I . Since there is no distinction between coordinates in the primal and the dual bases, we omit the $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ notations in this section. For all n -bit strings a, b , $S^{\otimes n} X^a Z^b (S^\dagger)^{\otimes n} = i^{a_1 + \dots + a_n \bmod 4} X^a Z^{a+b}$ and $H^{\otimes n} X^a Z^b H^{\otimes n} = (-1)^{a \cdot b} X^b Z^a$. Therefore, $S^{\otimes n}$ and $H^{\otimes n}$ respectively induce $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$.

The challenge of using a self-dual basis lies in the implementation of the unitary Π_r (field multiplication) that induces $\begin{pmatrix} r & 0 \\ 0 & r^{-1} \end{pmatrix}$. Fast multiplication methods with respect to a polynomial basis are known; however, no polynomial basis of $\text{GF}(2^n)$ is also self-dual if $n \geq 2$ [25]. Our solution is to use special self-dual bases that can be efficiently converted to and from polynomial bases. These special self-dual bases are constructed with *Gauss periods*, and are known for *admissible* n 's (see Definition 11 below). According to [37], there are *infinitely many* admissible n 's under the extended Riemann Hypothesis. Our implementation in this section is restricted to these values of n :

Definition 11. *A natural number n is called admissible if the following two conditions hold:*

- (1) $2n+1$ is prime
- (2) $\gcd(e, n) = 1$, where e is the index of the subgroup generated by 2 in \mathbb{Z}_{2n+1}^* .

In the above, \mathbb{Z}_{2n+1}^* denotes the multiplicative group of \mathbb{Z}_{2n+1} . Since \mathbb{Z}_{2n+1}^* has $2n$ elements, $e = \frac{2n}{|\langle 2 \rangle|}$.

In the remainder of this section, we first describe the procedure of finding a self-dual basis using Gauss periods, and briefly explain the efficient conversion between these two representations. Then we describe the implementation of Π_r .

Since, for admissible values of n , $2n + 1$ is prime, Fermat's Little Theorem implies $2^{2n} \equiv 1 \pmod{2n + 1}$. So $2n + 1$ divides $2^{2n} - 1$, which implies that there is a primitive $(2n + 1)$ -th root of unity $\beta \in \text{GF}(2^{2n})$. One way to get β is the following. Let ξ be a generator of the multiplicative group of $\text{GF}(2^{2n})$. Because $\xi^{2^{2n}-1} = 1$, we can take $\beta = \xi^{(2^{2n}-1)/(2n+1)}$. Consider the set

$$\mathcal{S} = \{\beta + \beta^{-1}, \beta^2 + \beta^{-2}, \dots, \beta^n + \beta^{-n}\}. \quad (49)$$

We first show that \mathcal{S} is a self-dual normal basis of $\text{GF}(2^n)$ over $\text{GF}(2)$ (as defined in Section 4.1). Then we show how to efficiently convert between \mathcal{S} and a polynomial basis.

First we show that for an admissible n , 2 and -1 generate \mathbb{Z}_{2n+1}^* (i.e., $\langle 2, -1 \rangle = \mathbb{Z}_{2n+1}^*$). A proof is given in [16], and it can be rephrased as follows. Let γ generate the cyclic group \mathbb{Z}_{2n+1}^* . If e is the index of $\langle 2 \rangle$ in \mathbb{Z}_{2n+1}^* , then $2 = \gamma^e$. Furthermore, $\gamma^n = -1$. Since $\gcd(e, n) = 1$, there are integers k_1, k_2 such that $1 = ek_1 + nk_2$ and therefore, $\gamma \in \langle 2, -1 \rangle$, so $\mathbb{Z}_{2n+1}^* = \langle 2, -1 \rangle$.

Our next step showing \mathcal{S} is a self-dual basis follows from [38]. Since $\mathbb{Z}_{2n+1}^* = \langle 2, -1 \rangle$, it follows that

$$\{2^0, -2^0, 2^1, -2^1, \dots, 2^{n-1}, -2^{n-1}\} \equiv \{1, -1, 2, -2, \dots, n, -n\} \pmod{2n+1}. \quad (50)$$

and we can reorder the elements of \mathcal{S} as

$$\{\beta^{2^0} + \beta^{-2^0}, \beta^{2^1} + \beta^{-2^1}, \dots, \beta^{2^{n-1}} + \beta^{-2^{n-1}}\}. \quad (51)$$

The set in Eq. (51) as a subset of $\text{GF}(2^n)$ is equal to $\{\alpha^{2^0}, \alpha^{2^1}, \dots, \alpha^{2^{n-1}}\}$ where $\alpha = \beta + \beta^{-1}$ is called a *Gauss period* of type $(n, 2)$ over $\text{GF}(2)$. It is easy to see that $\beta + \beta^{-1} \in \text{GF}(2^n)$, for one can verify that $(\beta + \beta^{-1})^{2^n} = \beta + \beta^{-1}$.

Finally, we need to show that \mathcal{S} is a basis. We invoke Theorem 3.1 in [16] which implies that α is a normal element in $\text{GF}(2^n)$ (generating a normal basis as defined in Section 4.1). Then, from Corollary 3.5 in [16], any normal basis of Gauss period of type $(n, 2)$ over $\text{GF}(2)$ is self-dual when $n > 2$, so, \mathcal{S} is self-dual, as claimed.

Next, we show how to efficiently convert between \mathcal{S} and a polynomial basis. We define a mapping from $\text{GF}(2^n)$ to $\{0, 1\}^{n+1}$ as follows. If $a \in \text{GF}(2^n)$, then $a' = [0, a_1, \dots, a_n]^T$, where $a = a_1(\beta + \beta^{-1}) + \dots + a_n(\beta^n + \beta^{-n})$. In other words, a' is the coordinate of a with respect to the spanning set $\{1, \beta + \beta^{-1}, \beta^2 + \beta^{-2}, \dots, \beta^n + \beta^{-n}\}$. Including the element 1 makes this spanning set not a basis, but significantly simplifies the conversion between the following two spanning sets:

$$\mathcal{S}' = \{1, \beta + \beta^{-1}, \beta^2 + \beta^{-2}, \dots, \beta^n + \beta^{-n}\}, \quad (52)$$

$$\mathcal{T} = \{1, \beta + \beta^{-1}, (\beta + \beta^{-1})^2, \dots, (\beta + \beta^{-1})^n\}. \quad (53)$$

Notice that the set \mathcal{T} arises from adding 1 to a polynomial basis. We call \mathcal{S}' a *self-dual spanning set* and \mathcal{T} a *polynomial spanning set*. The fact that \mathcal{T} is not a basis does not affect how we represent a field element as a polynomial based on \mathcal{T} , i.e., $a = \sum_{i=0}^n a_i(\beta + \beta^{-1})^i$, and fast multiplication of two polynomials of this form still works.

Let $s_i = \beta^i + \beta^{-i}$, $t_i = (\beta + \beta^{-1})^i$, and let s'_i and t'_i be the $(n + 1)$ -bit string output by the mapping defined earlier. We now describe the linear transformation L_{n+1} that maps s'_i to t'_i for

all i (by right multiplication). The transformation L_{n+1} is not unique. A simple choice for L_{n+1} is based on the binomial expansion $(\beta + \beta^{-1})^j = \sum_{i=0}^j \binom{j}{i} \beta^{j-2i}$. More precisely, for general k , we can choose L_k as

$$(L_k)_{i,j} = \begin{cases} 0 & \text{if } i > j \text{ or } j - i \text{ is odd,} \\ \binom{j}{(j-i)/2} \bmod 2 & \text{otherwise,} \end{cases} \quad (54)$$

where $0 \leq i, j < k$. The operation L_k can be reversed. L_k is upper-triangular with 1's on the diagonal, which implies $\det(L_k) = 1$, so L_k is invertible.

Finally, we will find a unitary \mathcal{L}_k that induces L_k . (More precisely, we are inducing the matrix with identical diagonal blocks that is the $(k-1) \times (k-1)$ submatrix of L_k with the first row and column omitted.) The unitary \mathcal{L}_n also induces a conversion from \mathcal{S}' to \mathcal{T} . In [38], the following theorem is proved.

Theorem 2 ([38]). *Right multiplying L_{n+1} (L_{n+1}^{-1} respectively) by the vector representation (a') of an element $a \in \text{GF}(2^n)$ described above can be done using $O(n \log n)$ operations (additions and multiplications) in $\text{GF}(2)$.*

From this theorem, an efficient (classical) circuit for L_{n+1} can be built with $O(n \log n)$ CNOT gates. The intuition is that L_{n+1} can be decomposed as a product of $O(\log n)$ matrices, each with $O(n)$ 1's. Since the linear transformation can be done with $\text{GF}(2)$ additions and multiplications, it can be implemented with CNOT gates. A circuit for L_{n+1}^{-1} can be obtained by running the circuit for L_{n+1} backwards.

Here we prove Theorem 2 with a different approach – a recursive construction that also requires $O(n \log n)$ CNOT gates. First consider L_k as defined in Eq. (54) where $k = 2^t$ is a power of 2. Taking $k = 8$ as an example,

$$L_8 = \left[\begin{array}{cccc|cccc} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right]. \quad (55)$$

We use two properties of L_k when $k = 2^t$ (see L_8 above for an illustration):

- (1) Each L_k consists of three non-zero blocks: two identical diagonal blocks which is $L_{k/2}$ and a block above the diagonal which we call $\Gamma_{k/2}$ (which is almost like $L_{k/2}$ turned upside down).²
- (2) The first row of $\Gamma_{k/2}$ contains only zero's. The $(i+2)^{\text{th}}$ row of $\Gamma_{k/2}$ is the $(\frac{k}{2} - i)^{\text{th}}$ row of $L_{k/2}$ (where $0 \leq i \leq k/2 - 2$).

We first explain why these two properties hold, as illustrated in Figure 3. Take the Pascal's triangle (mod 2) with k rows, and rotate the entries 90 degrees counter-clockwise. This gives the (nontrivial) (i, j) entries of L_k when $i \geq j$ and $i - j$ is even. The stated properties for L_k primarily

²Note here we use Γ for something different from the previous section.

come from the fact that Pascal's triangle (mod 2) with k rows consists of 4 triangles of $k/2$ rows, the middle one only has zero entries, and the other three are identical copies of Pascal's triangle (mod 2) with $k/2$ rows. Also, the triangle is always left-right symmetric. Proofs of these are readily obtained from Lucas' Theorem³ [14] (a more accessible proof can be found online at [31]).

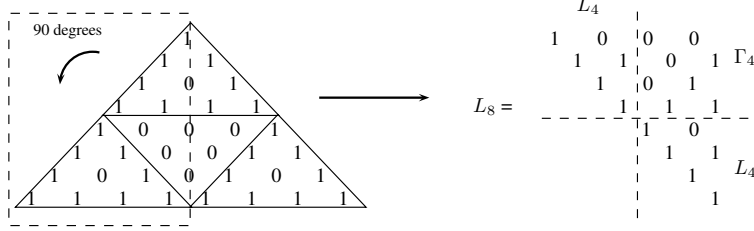


Figure 3: An illustration of the Pascal's triangle structure of the L_8 matrix. Taking the left half of an 8-level Pascal's triangle and rotating counter-clockwise by 90 degrees, we obtain the L_8 matrix. Note that the block Γ_4 is the horizontal reflection of the lower diagonal block L_4 with a downward shift, as described by property (2).

If we multiply L_k to a vector,

$$\left[\begin{array}{c|c} L_{k/2} & \Gamma_{k/2} \\ \hline 0 & L_{k/2} \end{array} \right] \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} L_{k/2} v_1 + \Gamma_{k/2} v_2 \\ L_{k/2} v_2 \end{bmatrix}. \quad (56)$$

Due to the relation between $\Gamma_{k/2}$ and $L_{k/2}$, the above mapping can be induced by the unitary \mathcal{L}_k implemented by the circuit in Figure 4. Using standard recursion analysis, the circuit contains $O(k \log k)$ CNOT gates.

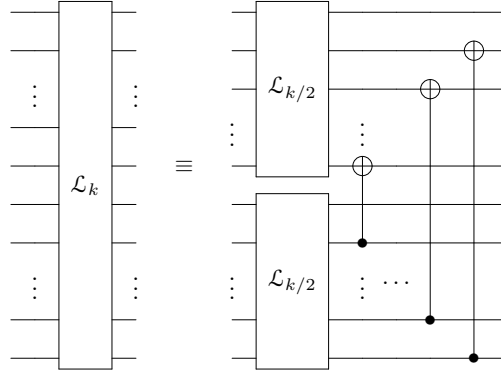


Figure 4: An example of representation conversion circuit which demonstrates the recursive structure.

For general values of k , let $t = \lceil \log_2 k \rceil$ and apply the above construction to obtain \mathcal{L}_{2^t} . We restrict the circuit for \mathcal{L}_{2^t} to a sub-circuit with the first k registers and the CNOT gates between them to obtain a circuit for \mathcal{L}_k that still has size $O(k \log k)$.

A circuit for \mathcal{L}_k^{-1} converting a vector from the self-dual representation to the polynomial representation can be obtained by running the circuit for \mathcal{L}_k backwards. The first qubit which corresponds to the additional “1” in \mathcal{S}' is always $|0\rangle$ and it remains untouched during the computation.

³Consider the base- p representation of integers m and n , where $m \geq n \geq 0$, and p is prime: $m = m_0 + m_1p + \dots + m_kp^k$, $n = n_0 + n_1p + \dots + n_kp^k$. Then, $\binom{m}{n} \equiv \binom{m_0}{n_0} \binom{m_1}{n_1} \dots \binom{m_k}{n_k} \pmod{p}$

Therefore, the first qubit can be safely removed in the circuit. It is kept in the analysis for conceptual simplicity.

Finally, we are ready to give the recipe for the fast multiplication of two elements $a, r \in \text{GF}(2^n)$ represented in the basis \mathcal{S}' :

1. Insert a zero at the beginning of the vector representations of a and r to get the vectors a' and r' with respect to the spanning set \mathcal{S}' .
2. Convert a' and r' to new representations \tilde{a} and \tilde{r} with respect to the polynomial spanning set \mathcal{T} , using the circuit for \mathcal{L}_{n+1}^{-1} .
3. Multiply \tilde{a} by \tilde{r} using Schönhage's multiplication algorithm [33] (denoted by $\tilde{\Pi}_r$ in figure 5). The result is a vector with respect to the polynomial spanning set $\{1, \beta + \beta^{-1}, (\beta + \beta^{-1})^2, \dots, (\beta + \beta^{-1})^{2n}\}$.
4. Apply the unitary \mathcal{L}_{2n+1} to the vector above so it is represented in the spanning set $\{1, \beta + \beta^{-1}, \beta^2 + \beta^{-2}, \dots, \beta^{2n} + \beta^{-2n}\}$. Then, discard the first element which is always 0. The result is the vector representation with respect to the spanning set $\{\beta + \beta^{-1}, \beta^2 + \beta^{-2}, \dots, \beta^{2n} + \beta^{-2n}\}$. Since β is the $(2n+1)$ -th root of unity in $\text{GF}(2^{2n})$ (i.e., $\beta^{2n+1} = 1$), we have $\beta + \beta^{-1} = \beta^{2n} + \beta^{-2n}$, $\beta^2 + \beta^{-2} = \beta^{2n-1} + \beta^{-2n+1}, \dots$. Therefore with n additional $\text{GF}(2)$ CNOTs, the resulting vector can be reduced to the one with respect to the permuted self-dual normal basis \mathcal{S} .

In Step 3, Schönhage's multiplication algorithm [33] uses a radix-3 FFT algorithm to do fast convolution. Readers not familiar with German may refer to [36] for another description of Schönhage's algorithm. This multiplication algorithm requires $O(n \log n \log \log n)$ operations (additions and multiplications). Additions can be implemented with CNOT gates. Multiplications involved in this radix-3 FFT are the ones between an element of the polynomial ring $\text{GF}(2)[x]/\langle x^{2m} + x^m + 1 \rangle$ (for certain m) and x (which is a $3m$ -th root of unity in $\text{GF}(2)[x]/\langle x^{2m} + x^m + 1 \rangle$). The result of this kind of multiplications is a shift of coefficients and it can be implemented by SWAP gates. Therefore, the whole multiplication method can be implemented with $O(n \log n \log \log n)$ CNOT gates. As an example, Figure 5 shows the implementation of Π_r in $\text{GF}(2^5)$.

It is easy to show that the radix-3 FFT algorithm has logarithmic depth: if the current step of this algorithm is working on a polynomial of degree k , in the next recursion step, it will work in parallel on three polynomials of degree $\lceil k/3 \rceil$. The total number of steps (i.e., the depth of the circuit) is therefore $O(\log n)$ for a polynomial of degree n . To multiply two polynomials of degree at most n , each recursion step essentially consists of three components: computing the radix-3 FFT, recursively doing $\lceil \sqrt{n} \rceil$ multiplications of polynomials of degree at most $\lceil \sqrt{n} \rceil$ (in parallel), and computing the inverse radix-3 FFT. Using a similar analysis, the depth of the polynomial multiplication circuit is $O(\log(n) + \log(n^{1/2}) + \log(n^{1/4}) + \dots + 1) = O(\log n)$. The logarithmic depth of the basis conversion circuit can be shown by its recursive structure (e.g., Figure 4). Therefore, the depth of the circuit for Π_r is $O(\log n)$.

The ancillary qubits can be reset to $|0\rangle$ using standard techniques in reversible computing. The result is a circuit for Π_r for any non-zero $r \in \text{GF}(2^n)$ with $O(n \log n \log \log n)$ CNOT gates.

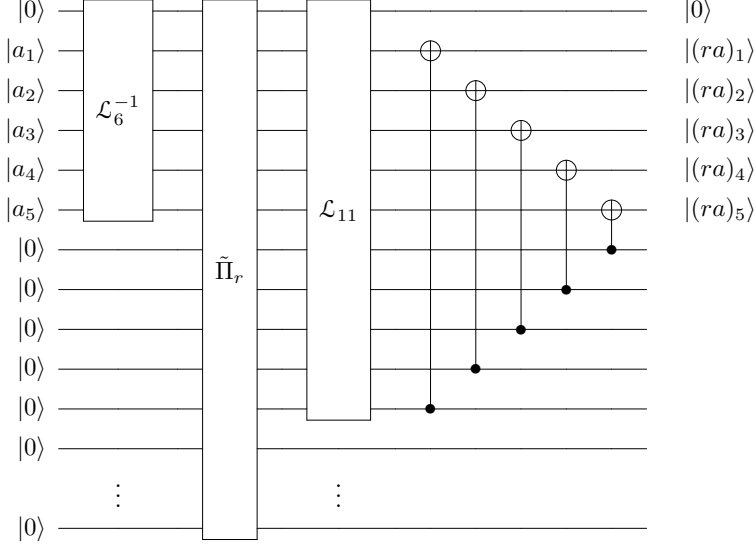


Figure 5: The implementation of Π_r for multiplication of a by r where $a, r \in \text{GF}(2^5)$. $\tilde{\Pi}_r$ is an implementation of Schönhage's multiplication algorithm. The input and output bits are with respect to a self-dual basis.

6 $\tilde{O}(n)$ implementations based on polynomial basis for $\text{GF}(2^n)$

In this section we present alternative circuit constructions for unitary 2-designs in terms of *polynomial* bases for $\text{GF}(2^n)$. The advantage of using polynomial bases is that the $\text{SL}_2(\text{GF}(2^n))$ generator $\begin{pmatrix} r & 0 \\ 0 & r^{-1} \end{pmatrix}$ for $r \neq 0$ is straightforward to implement⁴ with $O(n \log n \log \log n)$ Clifford gates with depth $O(\log n)$, as described at the end of Section 5.

For the generator, $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ we provide two different $\tilde{O}(n)$ circuit implementations in Subsections 6.1 and 6.2. However, we do not currently know how to implement the last generator $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ using only $\tilde{O}(n)$ gates. To circumvent this problem, we modify our ensemble for the unitary 2-design slightly. Instead of implementing every element of $\text{SL}_2(\text{GF}(2^n))$, we implement the elements that are lower triangular (i.e., $\Delta_2(\text{GF}(2^n))$), and we do this using $\tilde{O}(n)$ gates. This follows directly from combining the implementations for $\begin{pmatrix} r & 0 \\ 0 & r^{-1} \end{pmatrix}$ and $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ and by using Lemma 6. We can also implement all $M \in \nabla_2(\text{GF}(2^n))$ with respect to the dual basis (we denote this unitary \hat{U}_M), because with respect to the dual basis, the operations that induce $\begin{pmatrix} r & 0 \\ 0 & r^{-1} \end{pmatrix}$ and $\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ are $H^{\otimes n} \Pi_r H^{\otimes n}$ and $H^{\otimes n} \Gamma_V H^{\otimes n}$ (respectively). In Subsection 6.3 we show how to combine the implementations of $\Delta_2(\text{GF}(2^n))$ in the primal basis and $\nabla_2(\text{GF}(2^n))$ in the dual basis to achieve Pauli mixing. This results in an exact unitary 2-design with the desired complexity.

6.1 Implementation of $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ with $O(n \log n \log \log n)$ non-Clifford gates

Here we provide an implementation of $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ using $O(n \log n \log \log n)$ gates that can be organized so as to have depth $O(\log n)$. This construction uses non-Clifford gates but they compose to a Clifford unitary. (The next subsection contains a slightly less efficient construction using only

⁴ In this section, “implement a mapping” abridges “implement the unitary that induces a mapping according to Definitions 8 or 10” and so on.

Clifford gates.)

The operation that we need to implement is Γ_W , defined in Eqs. (36) and (37) (with V set to W). Recall that W is the primal-to-dual basis conversion matrix of Eq. (9). Since we are setting the primal basis to a polynomial basis, W is a *Hankel matrix*: for all j, k, j', k' , if $j + k = j' + k'$ then $W_{jk} = W_{j'k'}$. We make use of this property in this and the next subsection. From Eq. (36),

$$\Gamma_W|c\rangle = i^{\sum_{j=1}^n \sum_{k=1}^n W_{jk} c_j c_k} |c\rangle. \quad (57)$$

Note that it suffices to compute the exponent of i using mod 4 arithmetic, and the exponent has the form

$$\begin{bmatrix} c_1 & \cdots & c_n \end{bmatrix} W \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}. \quad (58)$$

This problem is related to the problem of computing convolutions. Recall that the *convolution* of two d -dimensional vectors u and v is defined as the $(2d - 1)$ -dimensional vector w such that

$$w_0 + w_1 T + w_2^2 T^2 + \cdots + w_{2d-2} T^{2d-2} \quad (59)$$

$$= \left(u_0 + u_1 T + v_2^2 T^2 + \cdots + u_{d-1} T^{d-1} \right) \left(v_0 + v_1 T + v_2^2 T^2 + \cdots + v_{d-1} T^{d-1} \right) \quad (60)$$

as polynomials over T . The product of a Hankel matrix with a vector reduces to convolution, as shown in the next proposition.

Proposition 3. *The product of an $n \times n$ Hankel matrix with an n -dimensional vector reduces to the problem of computing the convolution of two $(2n - 1)$ -dimensional vectors.*

Proof. This can be seen by comparing

$$\begin{bmatrix} x_1 & x_2 & \cdots & x_n \\ x_2 & x_3 & \cdots & x_{n+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_n & x_{n+1} & \cdots & x_{2n-1} \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad (61)$$

with the middle components of the convolution of $[x_1, \dots, x_{2n-1}]$ and $[0, \dots, 0, y_n, \dots, y_1]$. The convolution is a $(4n - 3)$ -dimensional vector that is the vector in Eq. (61) padded with $2n - 2$ components on the left and $n - 1$ components on the right. \square

Returning to the computation of Eq. (58), we can compute $e_1, \dots, e_n \in \mathbb{Z}_4$, given by

$$\begin{bmatrix} e_1 \\ \vdots \\ e_n \end{bmatrix} = W \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix}, \quad (62)$$

with a fast algorithm for polynomial multiplication⁵ over the ring \mathbb{Z}_4 using only $O(n \log n \log \log n)$ gates (see, for example, Theorem 8.23 in [36]). Then Eq. (58) for the exponent for i in Γ_W can be

⁵We conjecture that it is possible to slightly reduce the gate count for this construction from $O(n \log n \log \log n)$ to $(n \log n) 2^{O(\log^* n)}$ by employing the improved algorithms for integer multiplication initiated by Fürer [15].

obtained from the $2n$ ancillary qubits containing e_1, \dots, e_n (each e_j is a two-bit string) and the n qubits containing c_1, \dots, c_n as follows. For each $j \in \{1, \dots, n\}$, apply a controlled- Z gate between the high order bit of e_j and c_j and apply a controlled- S gate between the low order bit of e_j and c_j .

This construction explicitly uses the non-Clifford controlled- S gates, since the underlying ring is \mathbb{Z}_4 and addition mod 4 requires non-Clifford gates. The construction uses polynomial multiplications, so it follows from the circuit depth analysis in Section 5 that the circuit depth of this construction is $O(\log n)$. In the next subsection, we describe a different procedure for implementing $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ that is slightly less efficient, but uses only Clifford gates.

6.2 Implementation of $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ with $O(n \log^2 n \log \log n)$ Clifford gates

Here we provide an implementation of $\begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ using $O(n \log^2 n \log \log n)$ Clifford gates that can be organized so as to have depth $O(\log^2 n)$. In the previous subsection, the computation is reduced to a convolution in mod 4 arithmetic, and we needed non-Clifford gates to compute this efficiently. Here, we use a recursive procedure that is based on convolutions in mod 2 arithmetic, which can be performed efficiently with Clifford gates. We assume all notation from the previous subsection.

To simplify our presentation, we assume that n is a power of 2 (though our approach can be generalized to arbitrary n by dividing unevenly in the recursive step, as $n = \lfloor \frac{n}{2} \rfloor + \lceil \frac{n}{2} \rceil$). We divide W into four $\frac{n}{2} \times \frac{n}{2}$ blocks as

$$W = \begin{bmatrix} W^{(11)} & W^{(12)} \\ W^{(21)} & W^{(22)} \end{bmatrix} \quad (63)$$

where $W^{(11)}, W^{(12)}, W^{(21)}, W^{(22)}$ are $\frac{n}{2} \times \frac{n}{2}$ Hankel matrices and $W^{(12)} = W^{(21)}$. Define

$$A = \begin{bmatrix} 0 & W^{(12)} \\ W^{(21)} & 0 \end{bmatrix}, \quad B = \begin{bmatrix} W^{(11)} & 0 \\ 0 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 0 \\ 0 & W^{(22)} \end{bmatrix}. \quad (64)$$

Clearly,

$$\begin{bmatrix} I & 0 \\ W & I \end{bmatrix} = \begin{bmatrix} I & 0 \\ A & I \end{bmatrix} \begin{bmatrix} I & 0 \\ B & I \end{bmatrix} \begin{bmatrix} I & 0 \\ C & I \end{bmatrix} \quad (65)$$

so we can implement Γ_A , Γ_B , and Γ_C separately, and compose them to obtain Γ_W .

We first show how to implement Γ_A using $O(n \log n \log \log n)$ gates. From Eq. (37),

$$\Gamma_A |c\rangle = (-1)^{\sum_{j=1}^{n/2} \sum_{k=n/2+1}^n W_{jk} c_j c_k} |c\rangle. \quad (66)$$

The expression for the exponent of -1 above can be computed in mod 2 arithmetic, and has the form

$$\begin{bmatrix} c_1 & \dots & c_{\frac{n}{2}} \end{bmatrix} W^{(12)} \begin{bmatrix} c_{\frac{n}{2}+1} \\ \vdots \\ c_n \end{bmatrix}. \quad (67)$$

Once again, by Proposition 3, the above product of a Hankel matrix with a vector reduces to convolution, and hence polynomial multiplication over the field $\text{GF}(2)$. We can compute the bits

$e_{\frac{n}{2}+1}, \dots, e_n$, defined as

$$\begin{bmatrix} e_{\frac{n}{2}+1} \\ \vdots \\ e_n \end{bmatrix} = W^{(12)} \begin{bmatrix} c_{\frac{n}{2}+1} \\ \vdots \\ c_n \end{bmatrix} \quad (68)$$

in $\frac{n}{2}$ ancillary registers using only $O(n \log n \log \log n)$ gates. Moreover, since the convolution is with respect to entries of W —which are constants in our setting—all the gates can be Clifford gates (in fact, CNOT gates). Then we can apply $O(n)$ controlled- Z gates between the bits $e_{\frac{n}{2}+1}, \dots, e_n$ and $c_1, \dots, c_{\frac{n}{2}}$ (respectively) to apply the phase that correctly implements Γ_A .

What remains is to compute Γ_B and Γ_C . Each of these is equivalent to computing an instance of the original problem of size $n/2$. In the bottom of the recurrence (when W is a 1×1 matrix), a single S (phase) gate computes Γ_W . The gate cost $G(n)$ of the recursive procedure satisfies the recurrence

$$G(n) = 2G(n/2) + O(n \log n \log \log n), \quad (69)$$

whose solution satisfies

$$G(n) \in O(n \log^2 n \log \log n). \quad (70)$$

This recursive construction needs polynomial multiplication in each recursion step. According to the circuit depth analysis for polynomial multiplication in Section 5, the circuit depth is $O(\log n + \log \frac{n}{2} + \dots + 1) = O(\log^2 n)$.

6.3 Pauli mixing from $\Delta_2(\text{GF}(2^n))$ and $\nabla_2(\text{GF}(2^n))$ in different bases

Here, we show how to achieve Pauli mixing by implementing U_M for $M \in \Delta_2(\text{GF}(2^n))$ and \hat{U}_M for $M \in \nabla_2(\text{GF}(2^n))$. We will explain our approach in two parts. In the first part, we explain the actual generation and construction of the ensemble of unitaries—which is simple, but the resulting ensemble no longer corresponds to $\text{SL}_2(\text{GF}(2^n))$, so it is not clear that the ensemble is a unitary 2-design. In the second part, we prove that the new ensemble is Pauli mixing, so it is indeed a unitary 2-design.

The construction is based on the following decomposition of elements of $\text{SL}_2(\text{GF}(2^n))$, along the lines of Eq. (18):

$$\begin{pmatrix} \alpha & \gamma \\ \beta & \delta \end{pmatrix} = \begin{cases} \begin{pmatrix} 1 & 0 \\ \frac{\beta}{\alpha} & 1 \end{pmatrix} \begin{pmatrix} \alpha & \gamma \\ 0 & \alpha^{-1} \end{pmatrix} & \text{if } \alpha \neq 0 \\ \begin{pmatrix} \gamma & 0 \\ \delta & \gamma^{-1} \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} & \text{if } \alpha = 0. \end{cases} \quad (71)$$

Note that all matrices in this decomposition are lower triangular, upper triangular, or $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$. Lower triangular matrices can be implemented in the primal basis; upper triangular matrices can be implemented in the dual basis; and $\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ can be implemented in any self-dual basis (by $H^{\otimes n}$).

The procedure to generate an element of the ensemble is as follows.

Generation procedure:

1. Sample $\begin{pmatrix} \alpha & \gamma \\ \beta & \delta \end{pmatrix} \in \text{SL}_2(\text{GF}(2^n))$ according to the uniform distribution.
- 2.1 If $\alpha \neq 0$ then
 - set M_1 to $\begin{pmatrix} \alpha & \gamma \\ 0 & \alpha^{-1} \end{pmatrix}$
 - set M_2 to $\begin{pmatrix} 1 & 0 \\ \beta/\alpha & 1 \end{pmatrix}$
 - construct the Clifford group element $U_{M_2} \circ \widehat{U}_{M_1}$ (composition of two circuits).
- 2.2 Else if $\alpha = 0$ then
 - set M to $\begin{pmatrix} \gamma & 0 \\ \delta & \gamma^{-1} \end{pmatrix}$
 - construct the Clifford group element $U_M \circ H^{\otimes n}$ (composition of two circuits).

Note that the composition in step 2.1 is along the lines of the first case of Eq. (71) and the composition in step 2.2 is along the lines of the second case of Eq. (71). In each case, a Clifford group element with gate complexity $O(n \log n \log \log n)$ (or Clifford-gate complexity $O(n \log^2 n \log \log n)$) results; however, the subset of all Cliffords that can arise by this procedure does not have the structure of $\text{SL}_2(\text{GF}(2^n))$ because of the disparate coordinate systems being used for the components. This concludes the description of the generation and construction of elements of the ensemble.

We now explain why the ensemble resulting from the above procedure is a unitary 2-design in spite of the mismatched bases used to convert the matrices arising from Eq. (71) into Clifford unitaries. First, we consider the mixing property over the Paulis that results from U_M for a random $M \in \Delta_2(\text{GF}(2^n))$, and similarly for $\nabla_2(\text{GF}(2^n))$. Partition the non-zero elements of $\text{GF}(2^n) \times \text{GF}(2^n)$ into these two (disjoint) subsets:

$$R_1 = \left\{ \begin{pmatrix} a \\ b \end{pmatrix} \in \text{GF}(2^n) \times \text{GF}(2^n) : a = 0 \text{ and } b \neq 0 \right\} \quad (72)$$

$$R_2 = \left\{ \begin{pmatrix} a \\ b \end{pmatrix} \in \text{GF}(2^n) \times \text{GF}(2^n) : a \neq 0 \right\}. \quad (73)$$

It is straightforward to verify that a random element $M \in \Delta_2(\text{GF}(2^n))$ uniformly mixes within R_1 and it uniformly mixes within R_2 in the following sense.

Lemma 7. *Let $M \in \Delta_2(\text{GF}(2^n))$ be chosen uniformly at random. Then, for any $\begin{pmatrix} a \\ b \end{pmatrix} \in R_1$, the distribution $M \begin{pmatrix} a \\ b \end{pmatrix}$ is uniform over R_1 and, for any $\begin{pmatrix} a \\ b \end{pmatrix} \in R_2$, the distribution $M \begin{pmatrix} a \\ b \end{pmatrix}$ is uniform over R_2 .*

A similar result holds for $\nabla_2(\text{GF}(2^n))$ with a and b switched in the definitions of R_1 and R_2 (we omit the simple proof of this).

To illustrate the consequences of Lemma 7 on the Paulis, we can organize the n -qubit Paulis into rows and columns where $X^{[a]}Z^{[b]}$ is in column a and row b . We choose the first row and column to be labeled by $a = 0$ and $b = 0$ and call them the *zero row* and *zero column*. The relative ordering of the remaining rows and columns does not affect our discussion; they are collectively called the *nonzero rows* and the *nonzero columns*. Figure 6 shows such a layout for the $n = 2$ case where the identity Pauli is excluded.

Based on Lemma 7, conjugating by U_M for a uniformly distributed $M \in \Delta_2(\text{GF}(2^n))$ causes the zero column to mix uniformly and also the complement of the zero column (consisting of all the nonzero columns) to mix uniformly. We call this effect *lower-triangular Pauli mixing*. Schematically, this is illustrated in Figure 7. We can similarly define *upper-triangular Pauli mixing*, corresponding to a transposed version of Figure 7. Sampling $M \in \nabla_2(\text{GF}(2^n))$ and then constructing the Clifford unitary \widehat{U}_M achieves upper-triangular mixing.

	IX	XI	XX
IZ	IY	XZ	XY
ZI	ZX	YI	YX
ZZ	ZY	YZ	YY

Figure 6: A natural arrangement of all the non-trivial 2-qubit Paulis into rows and columns. Pauli mixing requires a uniform distribution on the 15 items.

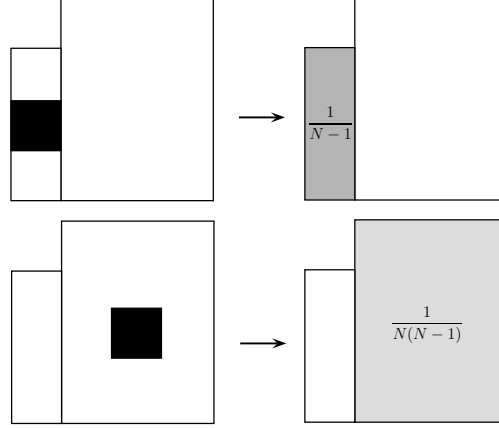


Figure 7: Illustration of lower-triangular Pauli mixing. Top: mixing effect within the zero column. Bottom: mixing effect within the complement of the zero column ($N = 2^n$).

We define one additional form of mixing, that we call *column Pauli mixing*, illustrated in Figure 8, where Paulis in the zero column do not change and any Pauli in a nonzero column mixes within its column. Such mixing is accomplished by choosing $M = \begin{pmatrix} 1 & 0 \\ \beta & 1 \end{pmatrix}$ for a uniformly random $\beta \in \text{GF}(2^n)$, and then constructing the Clifford unitary U_M .

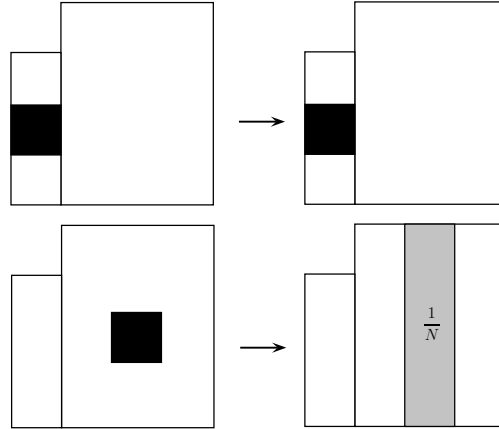


Figure 8: Illustration of column mixing. Top: elements in the zero column stay put. Bottom: elements in any nonzero column uniformly mix within the column ($N = 2^n$).

From Eq. (71), we can deduce that our procedure is applying a probabilistic mixture of the two procedures below. With probability $\frac{2^n}{2^n+1}$ it applies Procedure A; with probability $\frac{1}{2^n+1}$ it applies Procedure B ($\frac{1}{2^n+1}$ is the probability that $\alpha = 0$ for a random $\begin{pmatrix} \alpha & \gamma \\ \beta & \delta \end{pmatrix} \in \text{SL}_2(\text{GF}(2^n))$).

Procedure A:

1. Apply an upper-triangular Pauli mixing operation.
2. Apply a column Pauli mixing operation (independently from the first step).

Procedure B:

1. Apply $H^{\otimes n}$ (thereby transposing the layout of the Paulis).
2. Apply a lower-triangular mixing operation.

We now prove that the above mixture of Procedures A and B results in Pauli mixing.

Lemma 8. *The stochastic process of applying either Procedure A or Procedure B, with probabilities $\frac{2^n}{2^n+1}$ and $\frac{1}{2^n+1}$ (respectively) is Pauli mixing.*

Proof. For convenience, let $N = 2^n$. First, consider an initial Pauli in the zero row (i.e., $b = 0$ and it is of the form $X^{[a]}$ for some $a \neq 0$). Then, as illustrated in Figure 9, if Procedure A is

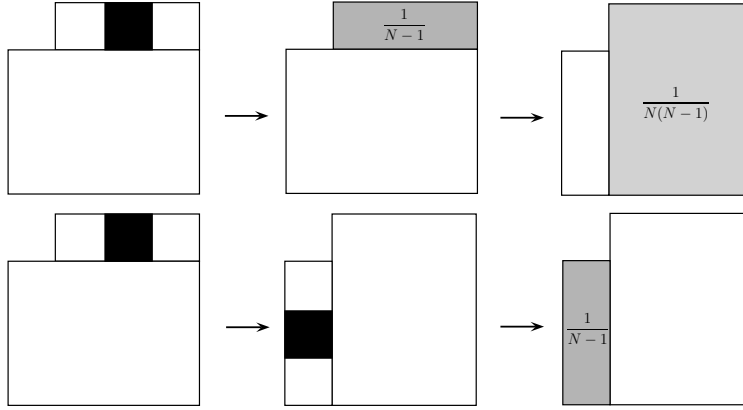


Figure 9: Illustration of mixing procedure starting in the zero row ($N = 2^n$). Top: Procedure A. Bottom: Procedure B.

applied, the result is a uniform distribution over all nonzero columns, where the probability of each Pauli is $\frac{1}{N(N-1)}$. On the other hand, if Procedure B is applied, the result is a uniform distribution on the zero column, where the probability of each Pauli is $\frac{1}{N-1}$. Consider the mixture of these distributions (Procedure A with probability $\frac{N}{N+1}$ and Procedure B with probability $\frac{1}{N+1}$). Since $\frac{N}{N+1} \frac{1}{N(N-1)} = \frac{1}{N^2-1}$ and $\frac{1}{N+1} \frac{1}{N-1} = \frac{1}{N^2-1}$, the result is the uniform distribution.

Next, consider the case of an initial Pauli that is not in the zero row (i.e., $X^{[a]}Z^{[b]}$ with $b \neq 0$). Then, as illustrated in Figure 10, if Procedure A is applied, the result is a two-level distribution: the probability of each Pauli in the zero column is $\frac{1}{N(N-1)}$; the probability of each Pauli in any nonzero column is $\frac{1}{N^2}$. On the other hand, if Procedure B is applied, the result is a uniform distribution over the nonzero columns, where the probability of each Pauli is $\frac{1}{N(N-1)}$. Consider the mixture of these distributions (Procedure A with probability $\frac{N}{N+1}$ and Procedure B with probability $\frac{1}{N+1}$). Since $\frac{N}{N+1} \frac{1}{N(N-1)} = \frac{1}{N^2-1}$ and $\frac{N}{N+1} \frac{1}{N^2} + \frac{1}{N+1} \frac{1}{N(N-1)} = \frac{1}{N^2-1}$, the result is the uniform distribution. \square

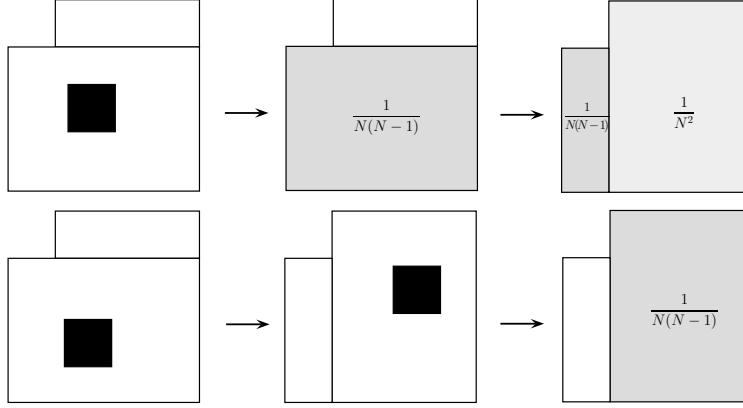


Figure 10: Illustration of mixing procedure starting in a nonzero row ($N = 2^n$). Top: Procedure A. Bottom: Procedure B.

7 Acknowledgments

We thank Olivia Di Matteo for pointing out reference [18] and Aram Harrow for pointing out reference [27]. We thank Joachim von zur Gathen, Mark Giesbrecht, and Arne Storjohann for discussions about finite field algorithms, and Daniel Gottesman and Aram Harrow for other discussions. This research was supported in part by Canada’s NSERC and CRC, a David R. Cheriton Graduate Scholarship, and the U.S. ARO.

References

- [1] S. Aaronson and D. Gottesman. Improved simulation of stabilizer circuits. *Phys. Rev. A*, 70:052328, Nov 2004. quant-ph/0406196.
- [2] C. H. Bennett, D. P. DiVincenzo, J. Smolin, and W. K. Wootters. Mixed-state entanglement and quantum error correction. *Phys. Rev. A*, 54:3824–3851, 1997.
- [3] W. Brown and O. Fawzi. Decoupling with random quantum circuits. *arXiv preprint arXiv:1307.0632*, 2013.
- [4] W. Brown, and O. Fawzi. Short random circuits define good quantum error correcting codes. In *Proceedings of the IEEE International Symposium on Information Theory, ISIT ’13*, pages 346–350.
- [5] J. L. Carter and M. N. Wegman. Universal classes of hash functions (extended abstract). In *Proceedings of the Ninth Annual ACM Symposium on Theory of Computing, STOC ’77*, pages 106–112, New York, NY, USA, 1977. ACM.
- [6] H. F. Chau. Unconditionally secure key distribution in higher dimensions by depolarization. *IEEE Trans. Inf. Theory*, 51(4):1451–1468, 2005. arXiv:quant-ph/0405016.
- [7] D. Gottesman (1997), *Stabilizer codes and quantum error correction*, PhD Thesis, Caltech arXiv:quant-ph/9705052.

- [8] D. Gottesman *A Theory of Fault-Tolerant Quantum Computation*, Phys. Rev. A, 57:127–137, 1998. quant-ph/9702029.
- [9] M.-D. Choi, Completely Positive Linear Maps on Complex Matrices. *Linear Algebra and Its Applications*, 10:285–290, 1975.
- [10] C. Dankert, R. Cleve, J. Emerson, and E. Livine. Exact and approximate unitary 2-designs and their application to fidelity estimation. *Phys. Rev. A*, 80:012304, Jul 2009.
- [11] I. Devetak. The private classical capacity and quantum capacity of a quantum channel. *IEEE Trans. Inf. Theory*, 51:44–55, 2005. arXiv:quant-ph/0304127.
- [12] D. P. DiVincenzo, D. W. Leung, and B. M. Terhal. Quantum data hiding. *IEEE Trans. Inf. Theory*, 48(3):580–598, 2002. arXiv:quant-ph/0103098.
- [13] J. Emerson, Y. S. Weinstein, M. Saraceno, S. Lloyd, D. G. Cory. Pseudo-random unitary operators for quantum information processing. *Science*, 302(5653):2098–2100, 2003.
- [14] N. Fine. Binomial coefficients modulo a prime. *American Mathematical Monthly*, 589–592, 1947.
- [15] M. Fürer. Faster integer multiplication. *SIAM Journal on Computing*, 39(3):979–1005, 2009.
- [16] S. Gao, J. von zur Gathen, D. Panario, and V. Shoup. Algorithms for exponentiation in finite fields. *Journal of Symbolic Computation*, 29(6):879–889, 2000.
- [17] D. Gottesman The Heisenberg Representation of Quantum Computers. In *Proceedings of the XXII International Colloquium on Group Theoretical Methods in Physics*, pages 32–43, 1999.
- [18] D. Gross, K. Audenaert, and J. Eisert. Evenly distributed unitaries: on the structure of unitary designs. *Journal of mathematical physics*, 48:052104, 2007.
- [19] A. W. Harrow and R. A. Low. Random quantum circuits are approximate 2-designs. *Communications in Mathematical Physics*, 291(1):257–302, 2009. arXiv:0802.1919.
- [20] M. Hastings. Randomizing quantum states: Constructions and applications. *Nature Physics*, 5:255, 2009. arXiv:0809.3972.
- [21] P. Hayden, M. Horodecki, J. Yard, and A. Winter. A decoupling approach to the quantum capacity. *Open Systems and Information Dynamics*, 15:7–19, 2008.
- [22] P. Hayden, D. Leung, P. W. Shor, and A. Winter. Randomizing quantum states: Constructions and applications. *Communications in Mathematical Physics*, 250(2):371–391, 2004. quant-ph/0307104.
- [23] P. Hayden, D. W. Leung, and A. Winter. Aspects of generic entanglement. *Communications in Mathematical Physics*, 265(1):95–117, 2006. quant-ph/0407049.
- [24] P. Hayden and A. Winter. Counterexamples to the maximal p -norm multiplicativity conjecture for all $p > 1$. *Communications in Mathematical Physics*, 284(1):263–280, 2008.
- [25] M. Hazewinkel. *Handbook of algebra*, volume 1. Elsevier, 1996.
- [26] D. Leung. Choi’s proof as a recipe for quantum process tomography. *Journal of Mathematical Physics*, 44(2):528–533, 2003. quant-ph/0201119.

- [27] R. Low. Pseudo-randomness and Learning in Quantum Computation. PhD Thesis, University of Bristol, UK, 2010. arXiv:1006.5227.
- [28] R. Lidl and H. Niederreiter. *Introduction to Finite Fields and Their Applications*. Cambridge University Press, Cambridge, revised edition, 1994.
- [29] S. Lloyd. Capacity of the noisy quantum channel. *Phys. Rev. A*, 55:1613–1622, 1997.
- [30] M. Nielsen and I. Chuang. *Quantum computation and quantum information*. Cambridge University Press, 2000.
- [31] L. Riddle. <http://ecademy.agnesscott.edu/~lriddle/ifs/siertri/Pascalmath.htm>
- [32] A. Roy and A. J. Scott. Unitary designs and codes. *Designs, codes and cryptography*, 53(1):13–31, 2009.
- [33] A. Schönhage. Schnelle multiplikation von polynomen über körpern der charakteristik 2. *Acta Informatica*, 7(4):395–398, 1977.
- [34] P. W. Shor. The quantum channel capacity and coherent information. lecture notes, MSRI Workshop on Quantum Computation, 2002. Available online at <http://www.msri.org/publications/ln/msri/2002/quantumcrypto/shor/1/>.
- [35] O. Szehr, F. Dupuis, M. Tomamichel, and R. Renner. Decoupling with unitary approximate two-designs. *New Journal of Physics*, 15(5):053022, 2013.
- [36] J. von zur Gathen and J. Gerhard. *Modern Computer Algebra*. Cambridge University Press, 1999.
- [37] J. von zur Gathen and F. Pappalardi. Density estimates related to Gauss periods. In *Cryptography and Computational Number Theory*, pages 33–41. Springer, 2001.
- [38] J. von zur Gathen, A. Shokrollahi, and J. Shokrollahi. Efficient multiplication using type 2 optimal normal bases. In *Arithmetic of Finite Fields*, pages 55–68. Springer, 2007.
- [39] J. Watrous. Naimark’s theorem; characterizations of channels. “Theory of Quantum Information” lecture notes, <https://cs.uwaterloo.ca/~watrous/LectureNotes.html>.

A Proof of Lemma 1 and Corollary 1

Lemma 1. *Let \mathcal{E} be any ensemble of unitaries in \mathbb{U}_N . Then, the following are equivalent:*

- (1) \mathcal{E} is degree-2 expectation preserving.
- (2) \mathcal{E} is 2-query indistinguishable.
- (3) \mathcal{E} implements the full bilateral twirl.
- (4) \mathcal{E} implements the full channel twirl.

Corollary 1. *For $\mathcal{E} = \{p_i, U_i\}_{i=1}^k$, let $\mathcal{E}^\dagger := \{p_i, U_i^\dagger\}_{i=1}^k$.*

- (a) \mathcal{E} implements the full bilateral twirl if and only if \mathcal{E}^\dagger does.
(b) \mathcal{E} implements the full channel twirl if and only if \mathcal{E}^\dagger does.

Proof. We will show, in order, $(1) \Rightarrow (2) \Rightarrow (3) \Rightarrow (1)$, Corollary 1(a), then, $(2) \Rightarrow (4) \Rightarrow (3)$, and finally Corollary 1(b).

$(1) \Rightarrow (2)$: Consider any distinguishing circuit \mathcal{C} making up to two queries of U or U^\dagger . Note that the output state $\eta_2(\mathcal{C}, U)$ is a product of matrices with at most two factors of U and two factors of U^\dagger . Thus, each entry of $\eta_2(\mathcal{C}, U)$ is a polynomial of degree at most 2 in the matrix elements of U and at most 2 in the complex conjugates of those matrix elements. By hypothesis, \mathcal{E} is degree-2 expectation preserving, thus the following holds entrywise:

$$\sum_{i=1}^k p_i \eta_2(\mathcal{C}, U_i) = \int d\mu(U) \eta_2(\mathcal{C}, U) \quad (74)$$

and \mathcal{E} is 2-query indistinguishable.

$(2) \Rightarrow (3)$: This follows from the definition that the bilateral twirl circuit is a special case of a 2-query distinguishing circuit \mathcal{C} .

$(3) \Rightarrow (1)$: Let $\{|j\rangle\}_{j=1}^N$ be a basis for \mathbb{C}^N . Suppose \mathcal{E} implements the full bilateral twirl, so, $\forall \rho$,

$$\sum_{i=1}^k p_i U_i \otimes U_i \rho U_i^\dagger \otimes U_i^\dagger = \int d\mu(U) U \otimes U \rho U^\dagger \otimes U^\dagger. \quad (75)$$

Since the density matrices span the complex Hilbert space of all possible square matrices of the same dimension, the above relation holds if we replace ρ by $|a_1\rangle\langle a_3| \otimes |a_2\rangle\langle a_4|$, for all $a_1, a_2, a_3, a_4 \in \{1, \dots, N\}$. Furthermore, we can left- and right-multiply the above equation by $\langle a_5| \otimes \langle a_6|$ and $|a_7\rangle \otimes |a_8\rangle$. This gives

$$\sum_{i=1}^k p_i \langle a_5| U_i |a_1\rangle \langle a_6| U_i |a_2\rangle \langle a_3| U_i^\dagger |a_7\rangle \langle a_4| U_i^\dagger |a_8\rangle = \int d\mu(U) \langle a_5| U |a_1\rangle \langle a_6| U |a_2\rangle \langle a_3| U^\dagger |a_7\rangle \langle a_4| U^\dagger |a_8\rangle.$$

Repeating the above for all possible a_1, \dots, a_8 and applying linearity implies Eq. (1) and that \mathcal{E} is degree-2 expectation preserving.

Corollary 1(a): From Definition 2, \mathcal{E} is 2-query indistinguishable iff \mathcal{E}^\dagger is. Thus, by the equivalence between (2) and (3), \mathcal{E} implements the full bilateral twirl if and only if \mathcal{E}^\dagger does.

$(2) \Rightarrow (4)$: This follows from the definition that the channel twirl circuit is a special case of a 2-query distinguishing circuit \mathcal{C} .

$(4) \Rightarrow (3)$: We provide a proof for the most general unitary 2-design here. Readers interested in the special (but common) case when the ensemble \mathcal{E} consists only of Clifford unitaries and $N = 2^n$ can consult Appendix B for a short proof.

We begin with some relevant concepts in quantum information. Let $|1\rangle, \dots, |N\rangle$ denote an orthonormal basis for \mathbb{C}^N , $\mathcal{B}(\mathbb{C}^N)$ denote the set of all bounded $N \times N$ matrices, and $\Phi = \sum_{l,j=1}^N |l\rangle\langle j| \otimes |l\rangle\langle j|$. Let \mathcal{I} denote the identity map on $\mathcal{B}(\mathbb{C}^N)$. For any linear map $\Theta : \mathcal{B}(\mathbb{C}^N) \rightarrow$

$\mathcal{B}(\mathbb{C}^N)$, denote the *Choi-matrix* of Θ by $J(\Theta) = (\Theta \otimes \mathcal{I})(\Phi) = \sum_{l,j=1}^N \Theta(|l\rangle\langle j|) \otimes |l\rangle\langle j|$ [9]. Θ is completely positive if and only if $J(\Theta)$ is positive semidefinite [9] (see also [26, 39]). A quantum channel is a linear, trace-preserving, and completely positive map.

Suppose for every quantum channel Λ , $\mathbb{E}_{\mathcal{E}}(\Lambda) = \mathbb{E}_{\mu}(\Lambda)$. Then, $J(\mathbb{E}_{\mathcal{E}}(\Lambda)) = J(\mathbb{E}_{\mu}(\Lambda))$. Rephrasing this equality using Eqs. (5) and (6), we have

$$\sum_{i=1}^k p_i (U_i^{\dagger} \otimes I) (\Lambda \otimes \mathcal{I})((U_i \otimes I) \Phi (U_i^{\dagger} \otimes I)) (U_i \otimes I) = \int d\mu(U) (U^{\dagger} \otimes I) (\Lambda \otimes \mathcal{I})((U \otimes I) \Phi (U^{\dagger} \otimes I)) (U \otimes I). \quad (76)$$

We transform each side of the above equation in 3 steps, turning the Choi matrix of the twirled channel into the bilateral twirl of an operator closely related to the Choi matrix of Λ . First, for the LHS of Eq. (76), we apply the transpose trick $(U_i \otimes I) \Phi (U_i^{\dagger} \otimes I) = (I \otimes U_i^T) \Phi (I \otimes U_i^*)$, where T and $*$ denote the transpose and the complex conjugate respectively. Second, we commute the conjugation by $(I \otimes U_i^T)$ with $\Lambda \otimes \mathcal{I}$. We apply similar manipulations on the RHS of Eq. (76). The equation becomes

$$\sum_{i=1}^k p_i (U_i^{\dagger} \otimes U_i^T) (\Lambda \otimes \mathcal{I})(\Phi) (U_i \otimes U_i^*) = \int d\mu(U) (U^{\dagger} \otimes U^T) (\Lambda \otimes \mathcal{I})(\Phi) (U \otimes U^*). \quad (77)$$

Third, we apply to Eq. (77) the *partial transpose* of the second system: for any $A_1, A_2 \in \mathcal{B}(\mathbb{C}^N)$, this linear map takes $A_1 \otimes A_2$ to $A_1 \otimes A_2^T$. In particular, the partial transpose of $(I \otimes U_i^T)(\Phi)(I \otimes U_i^*) = \sum_{l,j=1}^N |l\rangle\langle j| \otimes (U_i^T |l\rangle\langle j| U_i^*)$ is equal to $\sum_{l,j=1}^N |l\rangle\langle j| \otimes (U_i^{\dagger} |j\rangle\langle l| U_i) = (I \otimes U_i^{\dagger})(\chi)(I \otimes U_i)$ where $\chi = \sum_{l,j=1}^N |l\rangle\langle j| \otimes |j\rangle\langle l|$ is the swap operator on $\mathbb{C}^N \otimes \mathbb{C}^N$. Eq. (77) becomes

$$\sum_{i=1}^k p_i (U_i^{\dagger} \otimes U_i^{\dagger}) (\Lambda \otimes \mathcal{I})(\chi) (U_i \otimes U_i) = \int d\mu(U) (U^{\dagger} \otimes U^{\dagger}) (\Lambda \otimes \mathcal{I})(\chi) (U \otimes U) \quad (78)$$

which is equivalent to

$$\mathcal{T}_{\mathcal{E}^{\dagger}}((\Lambda \otimes \mathcal{I})(\chi)) = \mathcal{T}_{\mu}((\Lambda \otimes \mathcal{I})(\chi)). \quad (79)$$

(In the above, we have used the fact $d\mu(U^{\dagger}) = d\mu(U)$.) Altogether, the transpose trick, the commutation, and the partial transpose transform Eq. (76) concerning the equality of the Choi-matrices of the two channel twirls for Λ into Eq. (79) establishing the equality of the two bilateral twirls of the matrix $(\Lambda \otimes \mathcal{I})(\chi)$.

It remains to apply Eq. (79) to a set of carefully chosen Λ 's to show that $\mathcal{T}_{\mathcal{E}^{\dagger}}(A) = \mathcal{T}_{\mu}(A)$ for a basis $\{A\}$ of the input space. This will show that \mathcal{E}^{\dagger} implements the full bilateral twirl. By Corollary 1, \mathcal{E} also implements the full bilateral twirl and the proof will be completed.

We consider Λ 's with a specific form. Let \mathcal{R} be the completely randomizing map on $\mathcal{B}(\mathbb{C}^N)$, i.e., $\mathcal{R}(\rho) = (\text{Tr} \rho) I / N$ for all $\rho \in \mathcal{B}(\mathbb{C}^N)$. Note that $J(\mathcal{R}) = (I \otimes I) / N$. Consider any bounded linear map $\tilde{\Lambda}$ that is trace preserving and for which $J(\tilde{\Lambda})$ is Hermitian (the latter property is called hermiticity preserving). Then, for sufficiently small, positive, λ , $\Lambda = (1 - \lambda)\mathcal{R} + \lambda\tilde{\Lambda}$ has positive semidefinite Choi-matrix (because the Choi-matrix of \mathcal{R} is proportional to the identity), and is therefore completely positive. Furthermore, Λ is linear and trace-preserving. So, Λ is a quantum channel. When we apply Eq. (79) to such Λ 's, the \mathcal{R} terms cancel out (because $(\mathcal{R} \otimes \mathcal{I})(\chi) = (I \otimes I) / N$ which is invariant under either bilateral twirl). Therefore, Eq. (79) holds for all linear, trace and hermiticity preserving maps $\tilde{\Lambda}$ (which are easier to construct than quantum channels).

We are ready to show that $\mathcal{T}_{\mathcal{E}^{\dagger}}(A) = \mathcal{T}_{\mu}(A)$ for a basis $\{A\}$ of the input space. We take $A = H_l \otimes H_j$ where $\{H_l\}_{l=1}^{d^2}$ is a basis for $\mathcal{B}(\mathbb{C}^N)$ with the following additional properties:

- (1) Each H_l is Hermitian.
- (2) $H_1 = I/\sqrt{N}$.
- (3) $\text{Tr}(H_l H_j) = \delta_{lj}$. In particular, H_l is traceless for $l > 1$.
- (4) The swap operator has a simple representation in this basis,

$$\chi = \sum_{l=1}^{d^2} H_l \otimes H_l. \quad (80)$$

Such basis exists for all N . When $N = 2^n$, H_l can be taken to be proportional to the Pauli matrices (see Eq. (81) for the last condition). For general N , we show in Appendix C that the *generalized Gell-Mann matrices* can be used to construct such H_l 's.

We will verify that $\mathcal{T}_{\mathcal{E}^\dagger}(H_l \otimes H_j) = \mathcal{T}_\mu(H_l \otimes H_j)$ for all $1 \leq l, j \leq d^2$ by considering four cases. First, the equality is immediate for $l = j = 1$. Second, for each $1 < j \leq d$ consider $\tilde{\Lambda}_{1j}$ defined by $\tilde{\Lambda}_{1j}(H_1) = H_1 + H_j$, and $\tilde{\Lambda}_{1j}(H_l) = 0$ for all $l \neq 1$. $\tilde{\Lambda}_{1j}$ is trace-preserving since each H_l is traceless for $l > 1$. Furthermore, $(\tilde{\Lambda}_{1j} \otimes \mathcal{I})(\chi) = (H_1 + H_j) \otimes H_1$ and partial transposing the second system gives $J(\tilde{\Lambda}_{1j})$, which implies $\tilde{\Lambda}_{1j}$ is Hermitian. Therefore, we can apply Eq. (79) to $\tilde{\Lambda}_{1j}$ and conclude $\mathcal{T}_{\mathcal{E}^\dagger}(H_j \otimes H_1) = \mathcal{T}_\mu(H_j \otimes H_1)$. Third, because of the symmetry of the bilateral twirl, $\mathcal{T}_{\mathcal{E}^\dagger}(H_1 \otimes H_j) = \mathcal{T}_\mu(H_1 \otimes H_j)$. Fourth, let $1 < j \leq l \leq d$ and consider $\tilde{\Lambda}_{jl}$ such that $\tilde{\Lambda}_{jl}(H_1) = H_1$, $\tilde{\Lambda}_{jl}(H_j) = H_l$, and $\tilde{\Lambda}_{jl}(H_{j'}) = 0$ for all $j' \neq 1$ and $j' \neq j$. With arguments similar to the second case, $\mathcal{T}_{\mathcal{E}^\dagger}(H_l \otimes H_j) = \mathcal{T}_\mu(H_l \otimes H_j)$. This completes the proof.

Corollary 1(b): We have established the equivalence between (3) and (4), thus, by Corollary 1(a) \mathcal{E} implements the full channel twirl if and only if \mathcal{E}^\dagger does. □

B Short proof for (4) \Rightarrow (3) in Lemma 1

Here, we consider the special case when $\mathcal{E} = \{p_i, U_i\}$ is an ensemble with Clifford unitaries and $N = 2^n$. We will show that if \mathcal{E} implements the full channel twirl then it implements the full bilateral twirl.

The proof relies on several definitions in Section 3. We will show that if \mathcal{E} implements the full channel twirl then it is necessarily Pauli mixing, and the rest follow from Lemma 2. Consider an ensemble $\mathcal{E} = \{p_i, U_i\}$ with Clifford unitaries U_i such that $\mathbb{E}_{\mathcal{E}}(\Lambda) = \mathbb{E}_\mu(\Lambda)$ for all quantum channels Λ . Take an arbitrary Pauli matrix $P \in \mathcal{Q}_n$ with $P \neq I$ and an overall phase so that $P = P^\dagger$. Let $\Lambda(\rho) = P\rho P^\dagger$. On one hand, $\mathbb{E}_{\mathcal{E}}(\Lambda)(\rho) = \sum_{i=1}^k p_i (U_i^\dagger P U_i) \rho (U_i^\dagger P U_i)^\dagger$. On the other hand, $\mathbb{E}_\mu(\Lambda)(\rho) = (1 - \lambda)\rho + \frac{\lambda}{2^{2n}-1} \sum_{Q \in \mathcal{Q}_n \setminus \{I\}} Q\rho Q^\dagger$ for some $0 \leq \lambda \leq 1$. Note that for each i , U_i is in the Clifford group so $U_i^\dagger P U_i$ is a Pauli matrix. Thus, we have two Kraus representations for the same twirled channel, both with Kraus operators in the quotient Pauli group \mathcal{Q}_n , which is a basis for $2^n \times 2^n$ matrices over \mathbb{C} . Invoking Theorem 8.2 of [30] concerning the degrees of freedom over these Kraus operators, the i -th term of $\mathbb{E}_{\mathcal{E}}(\Lambda)$ can only contribute to Q in $\mathbb{E}_\mu(\Lambda)$ if and only if $U_i^\dagger P U_i$ is equivalent to Q in \mathcal{Q}_n (see Section 3). Finally, each $Q \neq I$ appears with equal weight in $\mathbb{E}_\mu(\Lambda)(\rho)$, thus the distribution $\{p_i, U_i^\dagger P U_i\}$ is uniform over $\mathcal{Q}_n \setminus \{I\}$.

C Construction of the basis $\{H_l\}$

We want $\{H_l\}_{l=1}^{d^2}$ to be a basis for $\mathcal{B}(\mathbb{C}^N)$ with the following additional properties:

- (1) Each H_l is Hermitian.
- (2) $H_1 = I/\sqrt{N}$.
- (3) $\text{Tr}(H_l H_j) = \delta_{lj}$. In particular, H_l is traceless for $l > 1$.
- (4) The swap operator $\chi = \sum_{l=1}^{d^2} H_l \otimes H_l$.

We use the *generalized Gell-Mann matrices* for the construction. Let $H_1 = I/\sqrt{N}$. For $l = 2, \dots, N$, let $H_l = D_l/\sqrt{l(l-1)}$ where D_l is a diagonal matrix with $(D_l)_{1,1} = \dots = (D_l)_{(l-1,l-1)} = 1$, $(D_l)_{l,l} = -(l-1)$, and $(D_l)_{j,j} = 0$ for $l+1 \leq j \leq d$. For $1 \leq j_1 < j_2 \leq d$, let $X_{j_1,j_2} = (|j_1\rangle\langle j_2| + |j_2\rangle\langle j_1|)/\sqrt{2}$, $Y_{j_1,j_2} = i(-|j_1\rangle\langle j_2| + |j_2\rangle\langle j_1|)/\sqrt{2}$. Let $\{H_{d+1}, \dots, H_{d^2}\} = \{X_{j_1,j_2}, Y_{j_1,j_2}\}_{1 \leq j_1 < j_2 \leq d}$ with any ordering. Then, $\{H_l\}_{l=1}^{d^2}$ span $\mathcal{B}(\mathbb{C}^N)$, each H_l is Hermitian, and $\text{Tr}(H_l H_j) = \delta_{lj}$. Finally, the expression for the swap operator χ can be verified by checking that each of the d^4 matrix entries on the RHS has the value given by the LHS. The verification involves routine arithmetic, each off-diagonal element involves only 2 terms, and the diagonal elements can be expressed as simple telescopic sums.

D Elementary proof that Pauli mixing implies a unitary 2-design

Lemma 2. *Let \mathcal{E} be an ensemble of Clifford unitaries and \mathcal{E}_Q be as defined in Section 3. If \mathcal{E} is Pauli mixing, then \mathcal{E}_Q implements the full bilateral twirl.*

Proof. The goal is to show that $\mathcal{T}_{\mathcal{E}_Q}(\rho) = \mathcal{T}_\mu(\rho)$ for all density matrices ρ . Note that both $\mathcal{T}_{\mathcal{E}_Q}$ and \mathcal{T}_μ are linear transformations on $2^{2n} \times 2^{2n}$ matrices. Therefore, it suffices to show that $\mathcal{T}_{\mathcal{E}_Q}$ and \mathcal{T}_μ act identically on a basis for these matrices. We consider a basis that contains the identity matrix I_{2n} and the swap operator χ_{2n} acting on $2n$ qubits, completed with matrices M trace orthonormal to I_{2n} and χ_{2n} (i.e., $\text{Tr}(I_{2n}M) = \text{Tr}(\chi_{2n}M) = 0$). We will prove the following three claims:

1. $\mathcal{T}_\mu(I_{2n}) = \mathcal{T}_{\mathcal{E}_Q}(I_{2n}) = I_{2n}$,
2. $\mathcal{T}_\mu(\chi_{2n}) = \mathcal{T}_{\mathcal{E}_Q}(\chi_{2n}) = \chi_{2n}$, and
3. if $\text{Tr}(I_{2n}M) = \text{Tr}(\chi_{2n}M) = 0$, then $\mathcal{T}_\mu(M) = \mathcal{T}_{\mathcal{E}_Q}(M) = \mathbf{0}$.

Recall from Eqs. (3) and (4) that

$$\begin{aligned} \mathcal{T}_\mu(\rho) &= \int d\mu(U) U \otimes U \rho U^\dagger \otimes U^\dagger \quad \text{and} \\ \mathcal{T}_{\mathcal{E}_Q}(\rho) &= \sum_{i,j} p_i 2^{-2n} (U_i R_j \otimes U_i R_j) \rho (R_j^\dagger U_i^\dagger \otimes R_j^\dagger U_i^\dagger). \end{aligned}$$

It follows that the first claim holds trivially. Furthermore, since $\chi_{2n}(A \otimes B)\chi_{2n} = B \otimes A$, or equivalently, $\chi_{2n}(A \otimes B) = (B \otimes A)\chi_{2n}$, the second claim follows.

To prove the third claim, it suffices to show $\mathcal{T}_{\mathcal{E}_Q}(M) = \mathbf{0}$. This is because, for any $2^{2n} \times 2^{2n}$ matrices \tilde{M} , $\mathcal{T}_\mu(\tilde{M}) = \mathcal{T}_\mu(\mathcal{T}_{\mathcal{E}_Q}(\tilde{M}))$. In turns, this is due to the fact that $\forall V \in \mathbb{U}_{2n}, \forall \tilde{M}, \mathcal{T}_\mu(\tilde{M}) =$

$\mathcal{T}_\mu(V \otimes V \tilde{M} V^\dagger \otimes V^\dagger)$; applying the last identity to each unitary in \mathcal{E}_Q and invoking linearity gives the desired result.

We now show that $\mathcal{T}_{\mathcal{E}_Q}(\tilde{M}) = \mathbf{0}$. We make a crucial observation that $\chi_2 = \frac{1}{2}(I \otimes I + X \otimes X + Y \otimes Y + Z \otimes Z)$, and thus

$$\chi_{2n} = \frac{1}{2^n} \sum_{R_l \in \mathcal{Q}_n} R_l \otimes R_l. \quad (81)$$

Now, we use the fact that \mathcal{Q}_n is a basis for $2^n \times 2^n$ matrices to write $M = \sum_{ab} \alpha_{ab} R_a \otimes R_b$ for some $\alpha_{ab} \in \mathbb{C}$. We take $R_0 = I_n \in \mathcal{Q}_n$, so, the two conditions on M can be rephrased as $\alpha_{00} = 0$ and $\sum_a \alpha_{aa} = 0$. By linearity, we focus on analyzing $\mathcal{T}_{\mathcal{E}_Q}(R_a \otimes R_b)$ for any $(a, b) \neq (0, 0)$. Note that

$$\mathcal{T}_{\mathcal{E}_Q}(R_a \otimes R_b) = \sum_i p_i (U_i \otimes U_i) \left[\sum_j 2^{-2n} (R_j \otimes R_j) (R_a \otimes R_b) (R_j^\dagger \otimes R_j^\dagger) \right] (U_i^\dagger \otimes U_i^\dagger). \quad (82)$$

If $a \neq b$, $\exists c$ such that R_c commutes with R_a and anticommutes with R_b . So,

$$\begin{aligned} & 2 \sum_j (R_j \otimes R_j) (R_a \otimes R_b) (R_j^\dagger \otimes R_j^\dagger) \\ &= \sum_j (R_j \otimes R_j) (R_a \otimes R_b) (R_j^\dagger \otimes R_j^\dagger) + \sum_j (R_j R_c \otimes R_j R_c) (R_a \otimes R_b) (R_c^\dagger R_j^\dagger \otimes R_c^\dagger R_j^\dagger) = \mathbf{0} \end{aligned}$$

and $\mathcal{T}_{\mathcal{E}_Q}(R_a \otimes R_b) = \mathbf{0}$. If $a = b$,

$$\sum_j 2^{-2n} (R_j \otimes R_j) (R_a \otimes R_a) (R_j^\dagger \otimes R_j^\dagger) = (R_a \otimes R_a). \quad (83)$$

Substituting the above into Eq. (82) and using the fact that \mathcal{E} is Pauli mixing, we obtain

$$\mathcal{T}_{\mathcal{E}_Q}(R_a \otimes R_a) = \frac{1}{2^{2n-1}} \sum_{R_j \in \mathcal{Q}_n \setminus \{I\}} R_j \otimes R_j = T \quad (84)$$

for a matrix T independent of a . Putting all the pieces together,

$$\mathcal{T}_{\mathcal{E}_Q}(M) = \sum_{ab} \alpha_{ab} \mathcal{T}_{\mathcal{E}_Q}(R_a \otimes R_b) = \sum_a \alpha_{aa} \mathcal{T}_{\mathcal{E}_Q}(R_a \otimes R_a) = \left(\sum_a \alpha_{aa} \right) T = \mathbf{0}. \quad (85)$$

□

E Lower bounds for size and depth of unitary 2-designs

Let $\mathcal{E} = \{p_i, U_i\}_{i=1}^k$ be any exact unitary 2-design on n qubits. We show that a high probability set of the unitaries have size $\Omega(n)$ and depth $\Omega(\log n)$, assuming a universal gate set consisting of 1- and 2-qubit gates. Both proofs invoke only Definition 3, and they apply to unitary 2-designs that approximate the exact operation under Definition 2 or 3 in the diamond norm.

Suppose the circuit for U_i acts nontrivially on s_i qubits. We will show that $\sum_{i=1}^k p_i s_i \geq n/2$, so, on average the circuit size is at least $n/2$. Since \mathcal{E} implements the full bilateral twirl, the quantum operation $\rho \rightarrow \sum_{i=1}^k p_i U_i \rho U_i^\dagger = \frac{I}{2^n}$ is the complete randomization map on n qubits. For

each j , consider the input $|0\rangle\langle 0|$ on the j th qubit. Since the output on the j th qubit is $\frac{I}{2}$, with probability at least $\frac{1}{2}$, it has been acted on by one of the U_i 's. Define a matrix with rows labeled by $i = 1, \dots, k$, and columns labeled by $j = 1, \dots, n$, and the (i, j) entry is p_i if U_i acts nontrivially on qubit j . The above argument implies that each column sums to at least $1/2$. Also, by definition, the i th row sums to $s_i p_i$. The total of the row sums is equal to the total of the column sums, so, $\sum_{i=1}^k p_i s_i \geq n/2$, as claimed. Furthermore, consider the set $\mathcal{S} = \{i : s_i < n/4\}$. If $\sum_{i \in \mathcal{S}} p_i > 2/3$, $\sum_{i=1}^k p_i s_i \not\geq n/2$, so, with probability at least $1/3$, the circuit has size at least $n/4$.

For the lower bound on the depth, consider the bilateral twirl $\mathcal{T}_{\mathcal{E}}$ applied to the matrix $Z \otimes I^{\otimes n-1} \otimes Z \otimes I^{\otimes n-1}$,

$$\mathcal{T}_{\mathcal{E}}(Z \otimes I^{\otimes n-1} \otimes Z \otimes I^{\otimes n-1}) = \sum_{i=1}^k p_i (U_i(Z \otimes I^{\otimes n-1})U_i^\dagger) \otimes (U_i(Z \otimes I^{\otimes n-1})U_i^\dagger). \quad (86)$$

Express each $U_i(Z \otimes I^{\otimes n-1})U_i^\dagger$ as a linear combination of Pauli matrices, and define the weight t_i to be the number of qubits that are acted on nontrivially by at least one of the terms. Since each gate interacts with at most two-qubits, if the depth of the circuit for U_i is d_i , then $d_i \geq \log t_i$. We now show that most $U_i(Z \otimes I^{\otimes n-1})U_i^\dagger$ have weight $t_i \geq n/2$.

From Appendix D,

$$\mathcal{T}_{\mathcal{E}}(Z \otimes I^{\otimes n-1} \otimes Z \otimes I^{\otimes n-1}) = \frac{1}{2^{2n-1}} \sum_{R_j \in \mathcal{Q}_n \setminus \{I\}} R_j \otimes R_j \quad (87)$$

The fraction of R_j 's with weight less than $n/2$ is equal to

$$4^{-n} \sum_{l=0}^{\lfloor n/2 \rfloor} \binom{n}{l} 3^l \leq 4^{-n} \sum_{l=0}^{\lfloor n/2 \rfloor} \binom{n}{l} 3^{n/2} \leq 4^{-n} \cdot \frac{1}{2} \cdot 2^n \cdot 3^{n/2} \approx 0.866^n.$$

Let $\mathcal{T} = \{i : t_i \geq n/2\}$. Then $\sum_{i \in \mathcal{T}} p_i \rightarrow 1$ but in particular $\sum_{i \in \mathcal{T}} p_i \geq 1/2$, because otherwise, the RHS of Eq. (86) and (87) cannot be equal.

F Pauli group permutations are uniquely induced

Lemma 9. *Suppose that unitaries U and V have the property that they induce the same permutation on the Pauli group so that, for all $a, b \in \{0, 1\}^n$,*

$$UX^a Z^b U^\dagger \equiv V X^a Z^b V^\dagger, \quad (88)$$

where \equiv means equal up to a global phase that can be a function of a and b . Then $V = UX^c Z^d$ for some $c, d \in \{0, 1\}^n$ (up to a global phase). (Here a and b are binary strings, as opposed to elements of $\text{GF}(2^n)$, so we do not require the notation $[a]$ and $[b]$ that occurs in other sections.)

Proof. Note that Eq. (88) is equivalent to

$$X^a Z^b (U^\dagger V) (X^a Z^b)^\dagger = \lambda_{a,b} U^\dagger V \quad (89)$$

for all $a, b \in \{0, 1\}^n$ where $\lambda_{a,b}$ is the global phase in Eq. (88). We can express $U^\dagger V$ as

$$U^\dagger V = \sum_{c,d \in \{0,1\}^n} \alpha_{c,d} X^c Z^d. \quad (90)$$

Recall that $X^a Z^b$ and $X^c Z^d$ either commute or anticommute, depending on the value of the symplectic inner product⁶ of (a, b) and (c, d) (they commute when $(a, b) \cdot (c, d) = 0$ and anticommute otherwise). Using this fact and substituting Eq. (90) into Eq. (89), we obtain

$$\sum_{c,d \in \{0,1\}^n} (-1)^{(a,b) \cdot (c,d)} \alpha_{c,d} X^c Z^d = \sum_{c,d \in \{0,1\}^n} \lambda_{a,b} \alpha_{c,d} X^c Z^d. \quad (91)$$

Since the Paulis $X^c Z^d$ are linearly independent, the coefficients must match.

We now show that at most one $\alpha_{c,d}$ can be nonzero. Suppose two are nonzero: $\alpha_{c_1,d_1} \neq 0 \neq \alpha_{c_2,d_2}$ for some $(c_1, d_1) \neq (c_2, d_2)$. Then there exists (a, b) such that $(a, b) \cdot (c_1, d_1) \neq (a, b) \cdot (c_2, d_2)$. Then, from Eq. (91), we can deduce that

$$(-1)^{(a,b) \cdot (c_1,d_1)} = \lambda_{a,b} = (-1)^{(a,b) \cdot (c_2,d_2)}, \quad (92)$$

which is a contradiction. Therefore there is a unique nonzero $\alpha_{c,d}$, which implies $V = \alpha_{c,d} U X^c Z^d$. \square

⁶The symplectic inner product is defined as $(a, b) \cdot (c, d) = (\oplus_{k=1}^n a_k d_k) \oplus (\oplus_{k=1}^n b_k c_k)$.